User Guide Version 6.2

Volume 2: Running MICRESS

“Resolution of partial differential equations is more about art than science“.

Apocryphal quotation from Numerical Recipes in Fortran

“2 + 2 = 4 except for large values of 2“

Anonymous

“42“

Douglas Adams

Edited by:

MICRESS group

Contents

1 How to use this manual 6

2 Introduction 7

3 Running MICRESS 9

3.1 The MICRESS® Background 9
3.2 The input file/driving file 11
3.2.1 Language settings 11
3.2.2 Geometry 12
3.2.3 Flags 13
3.2.4 Phase field data structure 15
3.2.5 Restart options 15
3.2.6 Name of output files 16
3.2.7 Selection of the outputs 17
3.2.8 Time input data 17
3.2.9 Phase data 19
3.2.10 Grain input 22
3.2.11 Data for further nucleation 25
3.2.12 Phase interaction data 32
3.2.13 Concentration data 36
3.2.14 Phase diagram – input data 39
3.2.15 Initial concentrations 45
3.2.16 Material data for fluid flow 47
3.2.17 Material data for elastic calculation 48
3.2.18 Parameters for latent heat and 1D temperature field 49
3.2.19 Boundary conditions 49
3.2.20 Other numerical parameters 56
3.2.21 Parallel execution parameter 57
3.3 How to create an input file? 58
3.4 Start the calculation 58
3.5 Check the output 59

4 Further Useful Information 61

4.1 Considerations about 2D simulation 61
4.2 Driving force 61
4.2.1 Driving force options 61
4.2.2 Averaging of the driving force 62
4.2.3 Driving force calculation using linearised phase diagrams 63

4.3 Outputs 64
4.3.1 List of available outputs 64
4.3.2 How to use the driving force output 67
4.4 Time input data 68
4.4.1 Automatic time stepping 68
4.5 Phase data 68
4.5.1 Anisotropic phase definition and crystallographic symmetry 68
4.5.2 Faceted growth 70
4.5.3 Definition of crystallographic orientations 70
4.5.4 Categorization 71
4.6 Grain input 73
4.6.1 Stabilisation and analytical curvature 73
4.6.2 Voronoi construction 73
4.7 Data for further nucleation 74
4.7.1 The special status of phase number 0 74
4.7.2 Nucleation models 74
4.7.3 Shield time and shield distance 76
4.7.4 “Diffuse” phase model for unresolved eutectic or eutectoid regions 76
4.7.5 Advanced grain manipulation using “add_to_grain” 76
4.8 Phase interaction data 77
4.8.1 Particle pinning (Zener) model 78
4.8.2 Solute drag model 79
4.8.3 Anisotropy of interfacial energies and mobilities 81
4.9 Concentration data 82
4.9.1 Initial concentration 82
4.10 Phase diagram – input data 83
4.10.1 Using linearised phase diagrams 83
4.10.2 Coupling to thermodynamic databases 83
4.11 Parameters for latent heat and 1D temperature field 85
4.12 Boundary conditions 86
4.12.1 Pressure condition for TQ 86
4.12.2 1d far field approximation 86
4.12.3 1d temperature field 87
4.12.4 Thermal boundary conditions 87
4.12.5 Periodic boundary conditions 89
4.12.6 Gradient boundary conditions 89
4.12.7 Cooling rate and temperature gradient 89
4.13 Numerical parameters 90
4.13.1 2D simulations of 3D structures? 90
4.13.2 Grid resolution and interface mobilities 90

4.13.3 Type of potential ... 92
4.13.4 iface and nTupel ... 93
4.13.5 Phase minimum phMin ... 94
4.14 Modelling solidification ... 95
4.14.1 Theoretical background on solidification ... 95
4.14.2 The Stefan problem of moving interfaces ... 96
4.14.3 The phase-field method ... 97
4.15 Using the flow solver ... 99
4.15.1 Activating the flow solver ... 99
4.15.2 Composition driven convection ... 99
4.15.3 Boundary conditions ... 99
4.15.4 Numerical parameters ... 100
4.15.5 Adjusting numerical parameters ... 102
4.15.6 Optimizing flow solver performance ... 103

5 How to simplify your life ... 105

5.1 How to define an initial microstructure ... 105
5.2 How to read /include a microstructure file ... 105
5.3 How to describe an anisotropic/faceted phase ... 106
5.4 How to include nucleation ... 106
5.5 How to choose between stabilisation and analytical curvature ... 107
5.6 How to include a further alloy element ... 108
5.7 How to add a further phase ... 108
5.8 How to create .ges-files for TC-coupled simulations (overview) ... 109
5.9 How to perform a TC-coupled simulation ... 109
5.10 How to go 3D ... 112
5.11 How to understand and model solidification ... 112
5.12 How to perform sub-grain modelling ... 113
5.13 How to handle grains touching the boundaries ... 113

6 Literature ... 114

1 How to use this manual

This manual is created to help using MICRESS® more efficiently saving processing time and effort.

The opening section, “Introduction“ presents the capabilities of the software as well as its wide range of application.

Chapter 3 “Running MICRESS“ goes along the structure of an input file required for a MICRESS® simulation. It offers a brief overview of all sections, flags and settings to be found in the input file. In addition to the brief explanation, you will also find some examples that will facilitate your work.

In chapter 4 “Further Useful Information“, topics are analysed in more detail. In some cases, the theoretical background of certain topics is discussed as well. Accordingly, chapter 5 provides more detailed information on some of the settings to be found in the input file.

Chapter 5 “How to simplify your life” is written in a “question-answer“ mode. Here, you will find target-oriented information on the wide range of applications of MICRESS®. Moreover, you will learn how to modify the input file to simulate e.g. recrystallisation, phase transformations, grain growth etc. In the answers to topical questions you will find all you need to know on that specific topic. This will help you to focus on your interests as well as to set-up your model for simulation fast and easily thus saving your time and effort.

The literature used in this work is listed in chapter 6.

2 Introduction

The software MICRESS® (MICRostructure Evolution Simulation Software) is developed for time- and space-resolved numerical simulations of solidification, grain growth, recrystallisation or solid state transformations in metallic alloys. MICRESS® covers phase evolution, solutal and thermal diffusion and transformation strain in the solid state. It enables the calculation of microstructure formation in time and space by solving the free boundary problem of moving phase boundaries.

Microstructure evolution is governed essentially by thermodynamic driving forces, diffusion and curvature. In case of multicomponent alloys, the required thermodynamic data can either be provided to MICRESS® in the form of locally linearised phase diagrams, or by direct coupling to thermodynamic data sets via a special TQ interface, developed in collaboration with Thermo Calc™ AB, Stockholm.

MICRESS® is based on the multiphase-field method which defines a phase-field parameter for each phase involved. The phase-field parameter describes the fraction of each phase as a continuous function of space and time. Each single grain is mapped to a distinct phase-field parameter and is treated as an individual phase. A set of coupled partial differential equations is formed which describes the evolution of the phase-field parameter, together with concentration, temperature, stress and flow fields. The total set of equations is solved explicitly by the finite difference method on a cubic grid.

2D and 3D simulations are possible. The size of the simulation domain, the number of grains, phases and components is restricted mainly by the available memory size and CPU speed.

Telling us suggestions for improvements on the manual or commenting on the manual is possible by writing an e-mail to documentation@micress.de.

MICRESS® handles:

- 1-, 2- and 3-dimensional calculation domains
- arbitrary number of components, phases and grains
- solid-solid and solid-liquid interactions
- anisotropy of grain boundaries, mobility and energy

MICRESS® supports:

- coupling to thermodynamic databases (via the TQ-interface of Thermo Calc™)

Chapter 2 Introduction

In MICRESS® User Guide Part 2: Running MICRESS you will find:

- a short introduction to the MICRESS® input file
- a targeted and user-oriented description of the data pre-processing for the simulation of recrystallisation, grain growth and phase transformation processes of multiphase, multigrain and multicomponent problems of vast industrial importance
- a structured overview of the input file depending on the targets of the simulation
- a more detailed review of some specific MICRESS® features

A description of the phase-field phenomenology and theoretical background can be found in MICRESS Vol. 0: Phenomenological Background. MICRESS Vol. I: MICRESS Installation provides information about the installation of the software and explains how to use the software with the help of an example. MICRESS Part III: Post-processing explains the possibilities the user has while treating output results. MICRESS Part IV: Examples contains a collection of examples for simulations performed on various topics.

3 Running MICRESS

3.1 The MICRESS® Background

The ultimate goal of the MICRESS® simulation software is to be able to follow the microstructure evolution throughout the whole lifespan of a material, starting from the melt and its development with time and temperature changes (see Figure 3.1).

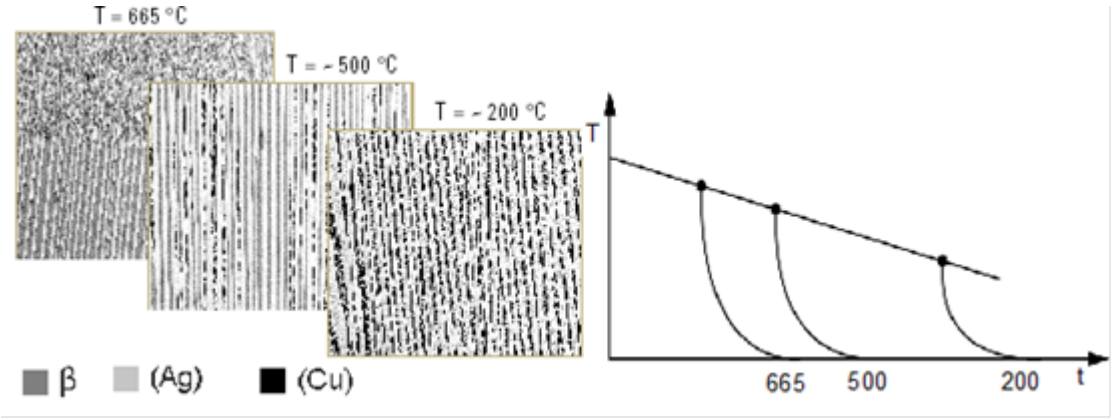


Figure 3.1 Microstructure is determined by the process (thermal history).

The numerical simulation of real processes such as microstructure evolution requires mathematical models that describe the existing physical ones. MICRESS® solves the so-called Stefan problem with a modified Gibbs-Thomson relation to model microstructure evolution. The solution of the mathematical problem of e.g. solidification involves process and material parameters, description of initial conditions, nucleation criteria, definition of the calculation domain as well as the consideration of some numerical parameters for PDE solvers. The phase-field method is used by MICRESS® as a numerical approach to the Stefan problem.

By now, the phase-field method is the most appropriate numerical approach for bridging the length scales between the interface capillarity length of a few nanometres and the millimetre scale of diffusion. The main characteristic feature of the phase-field model is the diffusiveness of the interface between two phases. The interface is described by a steep but continuous transition of the phase-field variable $\phi(x,t)$ between two states [14]. However, technical alloys consist of multiple grains, multiple phases and multiple components. Their description requires among others the introduction of multiple phase fields, description of multicomponent diffusion and coupling to thermodynamic databases. Some of the basic ideas of the multiphase-field approach (see Figure 3.2.) are:

- definition of one phase field for each phase and for each grain of this phase
- pairwise interaction for each pair of phases/grains similar to the standard phase field.
- possibility of implementation of specific phase/ grain boundary properties
- optional use of higher order interactions in triple or multiple junctions

Further concepts of the multiphase field approach are the coupling to thermodynamic databases and the coupling to mobility databases.

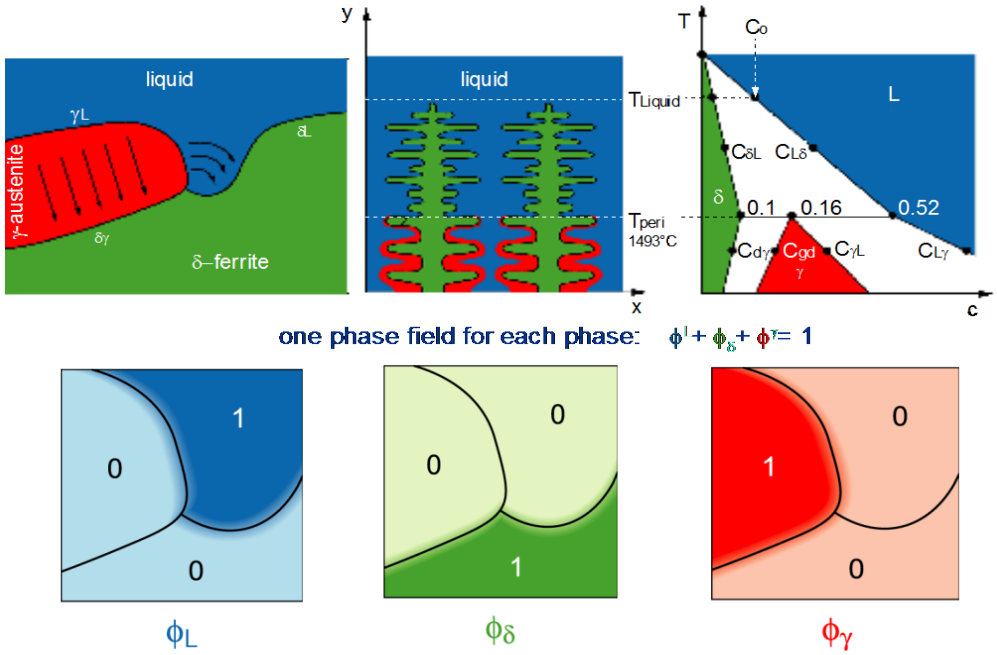


Figure 3.2 Multiphase-field model

In order to obtain reasonable simulation results, MICRESS® performs time loops among the nucleation model, the multiphase-field solver, multicomponent diffusion solver and the temperature solver. Coupling to thermodynamic database yields information about the nucleation undercooling, the driving force, the solute partitioning, the diffusion matrix, latent heat, etc. (see Figure 3.3). The obtained information is necessary for the different solvers in order to be able to perform the corresponding calculations for each time loop.

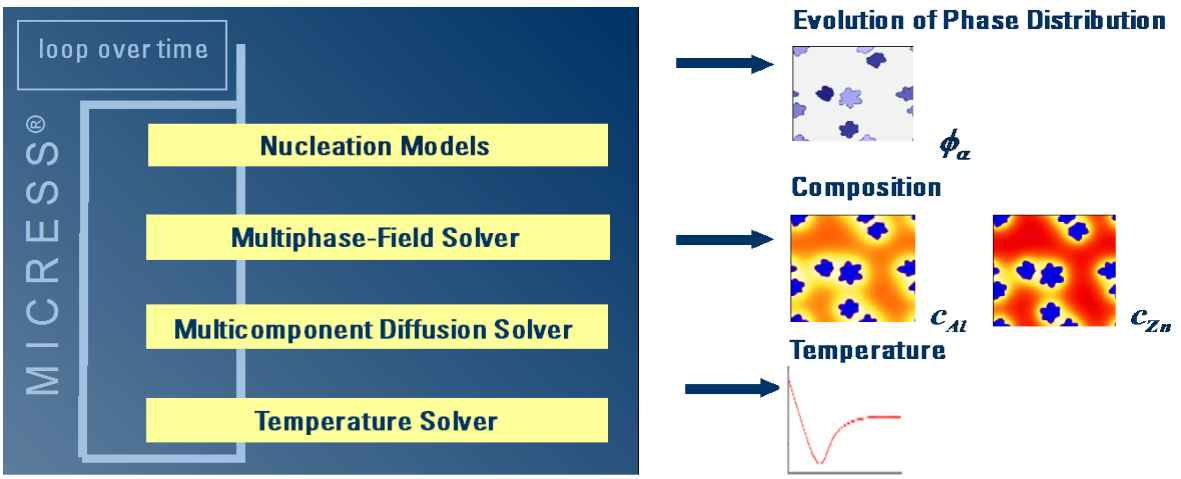


Figure 3.3 Basic structure of MICRESS

Figure 3.4 shows the solidified microstructure of an Al-Si-Mg-Cu-Ni alloy obtained in a MICRESS® simulation (left) and in a real process (left). It can be seen that both morphologies are quite similar, i.e. MICRESS® yields a good approximation of the solidification process.

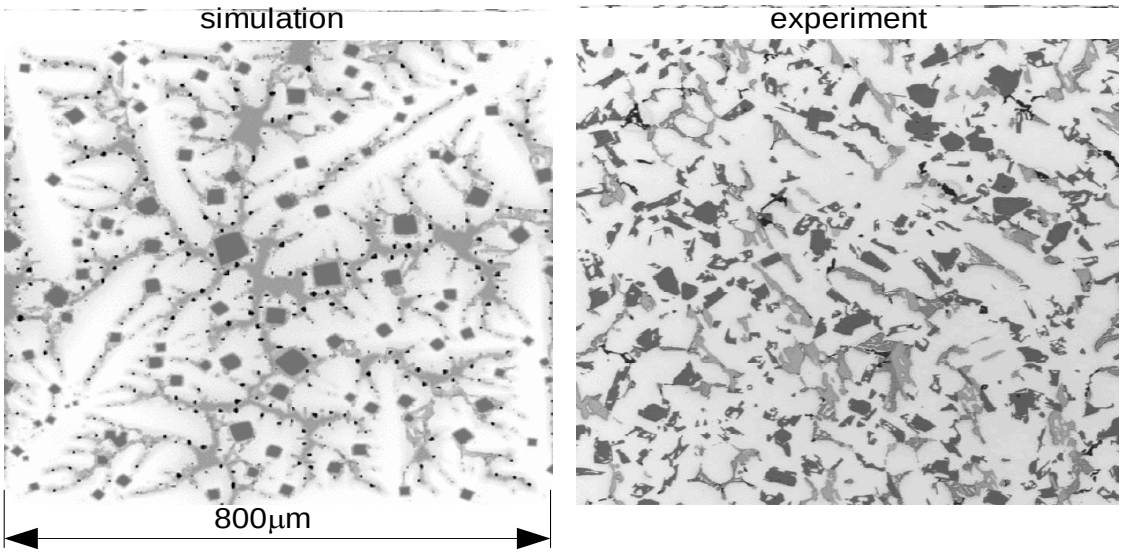


Figure 3.4 Solidified microstructure of Al-Si-Cu-Mg-Ni

3.2 The input file/driving file

MICRESS® requests input data from the terminal by a read statement. This input contains all the necessary information to start a simulation. It can be read from a text file, the so called driving file, via a shell or directly from keyboard. The driving file has the extension "*.dri" or "*_dri.txt". Some examples of driving files can be found in the examples directory of the installation directory. The MICRESS® input data is given in a sequential form with the input file divided into several sections. Their function and meaning will be explained in the following lines. The end of the chapter explains how to create an input file.

Important: The "driving file" ("*.dri" or "*_dri.txt") is your input file. You can add comments (by inserting # at the beginning of the line) that may be useful for your understanding of the file's contents. When running MICRESS® a new file with the extension ".in" or "_in.txt" is created which reflects the MICRESS® understanding of your inputs. All user comments are removed and a clean input file with standard notation is generated, which afterwards can be also used as a driving file!

3.2.1 Language settings

You can choose English, German and French language options. When running the input file, all text outputs are generated in the language selected by the user.

Driving File

```
…
# Language settings
# Please select a language:
#    'English', 'Deutsch' or 'Francais'
English
…
```

Example 3.1 Language settings

3.2.2 Geometry

The input begins with the dimension and the numerical resolution of the simulation domain, i.e. the user has to specify the number of numerical grid cells in each direction and the grid spacing. In this context, the user decides whether to perform a 1D, 2D or a 3D simulation. Here, the x, y and z direction are not completely equivalent: temperature gradients or profiles e.g. can only be defined in z-direction (except for temperature coupling, see below). Therefore, for 1D simulations, the number of cells in x and y direction ("AnzX" and "AnzY") should be set to 1. For 2D simulations, "AnzY" must be set to 1. The definition of a 2D calculation domain is shown in the driving file section below.

The grid spacing is specified in micrometers. It is one of the most important numerical parameters as it determines the numerical resolution.

Driving File

```
…
# Geometry
# -------------
# Grid size?
# (for 2D calculations: AnzY=1, for 1D calculations: AnzX=1, AnzY=1)
# AnzX:
300
# AnzY:
1
# AnzZ:
300
# Cell dimension (grid spacing in micrometers):
# (optionally followed by rescaling factor for the output
#   in the form of '3/4')
2.0000
…
```

Example 3.2 Geometry settings for a 2D simulation

Figure 3.5 MICRESS simulations in 3D

The grid resolution should be high enough to resolve the diffusion profiles, depending on the diffusion coefficients and the growth velocity of the interface, and the curvatures of the finest expected microstructures.

As an optional parameter, a rescaling factor for the graphical outputs can be given. Then, the outputs are written with a correspondingly smaller resolution. The simulation itself is not affected by this parameter.

3.2.3 Flags

Type of coupling

In the “Flags“ section the type of coupling to be used for the numerical approach has to be specified. The option “phase“ means that the pure phase-field model will be used with no coupling to other fields. The choice is best e.g. for grain growth simulations in pure, polycrystalline phases.

Driving File

```
…
# Flags
# --------
# Type of coupling?
# Options: phase concentration temperature temp_cyl_coord
#   [stress] [stress_coupled] [flow] [flow_coarse] [dislocation]
phase
…
```

Example 3.3 Type of coupling: phase

If “concentration“ is selected, concentration-field coupling will be performed. This option is the most common used for the analysis of alloys. In addition, [stress] or [stress_coupled] can be specified as optional parameters. Then, the calculation of the elastic stress field is activated and, if “stress_coupled“ is selected, the effect of stress on the driving force for phase transformation is also included. The latter option requires stress calculation in all numerical time steps and therefore is strongly increasing the simulation time.

The [flow…] options activate the MICRESS® flow module in the cases of phase- and concentration coupling. If [flow_coarse] is chosen a grid coarsening factor will be asked.

[stress] and [stress coupled] require a license for the MICRESS® elastic module.

[dislocation] parameter must be specified in order to activate the new recrystallization (RX) model based on a local dislocation density field read from a specific VTK file and coming, for example, from a Crystal Plasticity-Finite Element analysis (CP-FEM) of the forming process.

Important: The concentration coupling flag is not reasonable for the simulation of pure metal solidification!

The [temp_cyl_coord] is a special option which assumes a 2D or 3D temperature field in a cylindrical coordinate system.

Driving File

```
…
# Flags
# --------
# Type of coupling?
# Options: phase concentration temperature temp_cyl_coord
#   [stress] [stress_coupled] [flow] [flow_coarse] [dislocation]
concentration
# Type of potential?
# Options: double_obstacle multi_obstacle [fd_correction]
double_obstacle fd_correction
# Enable one dimensional far field approximation for diffusion?
# Options:   1d_far_field   no_1d_far_field
1d_far_field
# Number of cells for the 1D external field? (set to 0 for automatic)
500
# Shall an additional 1D field be defined in z direction
# for temperature coupling?
# Options: no_1d_temp 1d_temp 1d_temp_cylinder
#          1d_temp_polar [kin. Coeff]
# kin. Coeff: Kinetics of latent heat release (default is 0.01)
1d_temp
# Number of cells?
500
# cell width (micrometer):
100.000000000000
…
```

Example 3.4 Type of coupling: concentration with 1D_far_field and 1D_temp_field

Type of potential

The user can select either “double_obstacle“ or “multi_obstacle.

The ‘double_obstacle’ functional is of parabolic type: $g(\phi)=\phi(1-\phi)$, limited by discontinuities at $\phi=0$ and $\phi=1$. The ‘multi_obstacle’ functional [16] represents its generalized formulation for multiphase systems comprising contributions $g_{\alpha\beta}(\phi)=\phi_\alpha\,\phi_\beta$ for all pairwise phase interactions. In the special case of a two pha-se systems, it exactly recovers the ‘double_obstacle’

functional. The option ‘multi_obstacle’ may become default in future and is recommended to be used as a standard for all applications.

If ‘double_obstacle’ is selected for simulation of a multiphase problem, the anti-symmetric approximation [17] will be used. In this simplifying treatment, pairwise interactions between phases α and β are directly derived from the corresponding two phase free energy functionals $f_{\alpha\beta}$ instead of from the common free energy functional, thus neglecting higher order contributions $\alpha\beta\gamma$. As a consequence all dihedral angles at a junction will evolve as equal (i.e. a triple junction in 2-D will always exhibit three 120° angles) independent of the energies of the adjacent interfaces. Therefore, this approximation can only be used in cases in which all boundaries energies are equal or in which dihedral angles are of no importance. For a correct reproduction of dihedral angles in multiphase problems select ‘multi_obstacle’. Keep in mind that strongly differing interfacial energies may lead to high junction forces and extreme angles.

Option: “fd_correction“

The numerical solution of the phase-field equation based on finite differences (FD) inherently implies a discretization error which may strongly bias the simulation results. Usually, it results in too low interface velocities. The error is inversely correlated to the number of numerical cells resolving the diffuse interface profile. However, high numbers of interface cells are detrimental for computational performance and also favor undesired numerical interface trapping. The problem can be overcome by an improved numerical solution of the phase-field equation, which takes advantage of the a priori knowledge of the phase-field profile, determined by the choice of the potential function ([18], [19]). To activate this improved numerical solution, select “fd_correction“ and choose an interface width of 3-4 numerical cells. If you want to rerun a diffusion-controlled simulation which has been run before without the fd_correction option, it might be necessary to recalibrate the interface mobility. The option ‘fd_correction’ is recommended to be used as a standard for all applications and may become default in coming versions. It is however not yet compatible with temperature coupling.

The option “multi_obstacle“ especially comprises corrections for triple junction terms and is important for correct wetting characteristics.

“1d_far_field“

In case of “concentration“ coupling, a one-dimensional far field approximation for solute diffusion can be applied. This option is typically used for simulation of directional solidification and has two effects. The first is that in the upper part of the simulation domain diffusion is solved in 1D-approximation, starting from a given distance from a growing front. This distance is defined in the „Initial concentration“ input. The position of the growing front is defined as the point with the highest z-coordinate which contains phases other than the matrix phase (phase 0). The advantage of this approximation is a speed-up of the simulation run. The second effect is that an additional 1D concentration field is added attached at the top of the simulation domain. This is helpful for resolving long-range diffusion fields without having to increase the calculation domain. If “1d_far_field“ is enabled, one has to specify the length (in cells) of this “1D external extension“. The resolution of the external 1D-concentration field is the same as for the normal simulation domain. In addition, extra output files showing the concentration profiles outside the domain will be generated (“1D_Ext“ files).

“1D_temp“

If concentration coupling is activated, an explicit 1D temperature field can optionally be defined in z-direction. This field can be used as an improved thermal boundary condition for the microsimulation domain and thus

replaces the normal definition of thermal boundary conditions (see later in this chapter). In the 1D temperature field, heat flow and release of latent heat is solved explicitly.

The use of the 1D_temp option requires further inputs for the field size (number of cells) and the field resolution (cell width in micrometer) as well as further input of thermal diffusivities (parameter for latent heat) and boundary conditions. The total size of the temperature field must be equal or larger than the height of the microstructure simulation domain (in z-direction). As temperature is solved explicitly in the 1D temperature field, the use of latent heat is mandatory.

3.2.4 Phase field data structure

In this section, the initial dimensions for the internal fields iFace and nTupel must be specified. During runtime, the size of these fields is determined automatically, so in most cases the given values are of minor importance

The values have to be given relative to the size of the simulation domain, a value of 1.0 for iFace for example would assume that the whole calculation domain could be covered by (two-grain or two-phase) interfaces without exceeding the given initial list size. The same holds for nTupel and the coverage of the domain with triple or higher junctions. The actual usage for both fields can be found in the TabL output.

Driving File

```
...
# Phase field data structure
# ----------------------------------
# Coefficient for initial dimension of field iFace
#  [minimum usage] [target usage]
0.10
# Coefficient for initial dimension of field nTupel
#  [minimum usage] [target usage]
0.10
...
```

Example 3.5 iFace and nTupel

The arrays iFace and nTupel are fully dynamic, so the influence of the initial values is quite limited. In extreme cases, the specification of excessively high initial values can lead to a memory overflow during the program startup. Too low values can lead to an unnecessarily high number of reallocation steps which slow down the initialisation process e. g. in the case of grain growth simulation with a high number of initial grains. Initial values of 0.1 for both parameters are recommended and work in practically all cases.

Additional optional parameters can be specified for the “minimal“ and the “target“ usage. These parameters allow a fine-tuning of the reallocation criteria during runtime what can be helpful if a temporal memory overflow must be avoided. More details can be found or requested in the MICRESS® forum (www.micress.de).

3.2.5 Restart options

MICRESS® allows the user either to start a new simulation (“new“) or to restart from the last output of an old one (“restart“). The “restart“ option thus gives the possibility to continue a stopped simulation or to start various simulations (e.g. for a parameter variation) from a common starting point defined by a previous simulation. Thus, calculation time and effort can be saved.

Note: Even if the restart option is used, all input

Driving File

```
...
# Restart options
# ===============
# Restart using old results?
# Options:   new      restart [reset_time | structure_only]
restart
# Name of restart file?
aRestartFile
...
```

Example 3.6 Restart settings

parameters have to be specified like for a new simulation run. Those parameters which represent initial conditions like initial composition and temperature are replaced automatically by the values from the restart file.

Important: Extreme caution should be exercised when modifying parameters in conjunction with the restart feature. Several parameters like the size of the simulation domain (in cells), the number of used elements and phases, the number of diffusion contributions or nucleation types cannot be changed due to compatibility reasons. Which parameters can be changed depends also on the optional keyword which is used in the same line with "restart" (see below). Furthermore, the restart file must have been created using the same MICRESS® version and the same computer type.

If the restart option is used, the path and name of the restart file (without extension) has to be given in the next line. The restart file can be taken from a previous simulation. It has the extension .rest and is automatically created for the latest output time step if the corresponding restart output option is specified (see below).

The optional keyword "reset_time" in the same line after „restart" allows resetting the simulation time to 0. This is convenient for simulation of multiple process steps because e.g. the temperature-time history needs not to be repeated through all driving files for the different process steps and the initial temperature can be explicitly given in case of restart. Furthermore, all internal timers (e.g. for relinearisation or nucleation checks in certain intervals or internal diffusion time stepping) are reinitialized. This has the positive side effect that the number of diffusion contributions or nucleation types can now be modified!

Driving File

```
…
# Restart options
# ===============
# Restart using old results?
# Options:   new      restart [reset_time | structure_only]
restart structure_only
# How many restart files shall be read?
2
# For each restart file a grain number and (optionally) a
# shift in all dimensions (in grid cells) must be specified:
# grain number [shift X (int)  shift Y (int)  shift Z (int)] ?
3 10 0 65
# Name of restart file?
Results/Restart_1
# For each restart file a grain number and (optionally) a
# shift in all dimensions (in grid cells) must be specified:
# grain number [shift X (int)  shift Y (int)  shift Z (int)] ?
2 10 0 0
# Name of restart file?
Results/Restart_2
#
…
```

Example 3.7 Combining different microstructures using the "restart structure_only" option

The optional keyword "structure_only" in the same line after „restart" allows reading only the pure microstructure data without any type of numerical or status variables. This option not only permits the user to change nearly all numerical parameters, but also allows to alter geometry and resolution, and even to combine several microstructures from different restart files in one single simulation! For high flexibility in locating the individual microstructures, the content of the .rest files are positioned inside existing grains (with a specific grain number), which may have been created using the initial grain input, reading in initial microstructures from file, or even by preceeding input from a restart file. The size of the geometry which was used in the simulation creating the .rest file to be used here is required to exceed the size of the grain to be filled. A shift in the x, y, and z-direction (relative to the lower left corner) can be specified optionally. In case of symmetric or periodic microstructures (i.e. depending on the symmetry which was used in the simulation which created the .rest file), mirroring or periodic continuation is applied automatically.

3.2.6 Name of output files

After the restart options, a path and name for the simulation results must be given. This file name will be used for all output data files generated by MICRESS®.

In Example 3.8, the base output file name is “ReX3D“.

All output files will have the name “Rex3D“ with different file extensions. The different MICRESS® output files and the corresponding extensions are explained in chapter 4.3 and in the MICRESS manual Vol.3 on post processing.

Three modes for writing outputs are available among which the user has to make a choice in the next line: In “overwrite“ mode, already existing results with the same name will be overwritten without warning (if the operating system allows the user to do that). With “write_protected“, execution of the program will be stopped in this case and a warning is issued.

Driving File

```
...
# Name of output files
# ====================
# Name of result files?
Results/ReX3D
# Overwrite files with the same name?
# Options:  overwrite  write_protected  append
#           [zipped|not_zipped|vtk]
#           [unix|windows|non_native]
overwrite vtk
# VTK output file format options?
# Options:  binary_unzip  binary_zip  ascii
binary_unzip
...
```

Example 3.8 Output file names

The “append“ option, finally, allows to append the new simulation results to already existing files. This is especially helpful if the restart option is used. A further optional keyword allows changing the file format for example from the native binary format to a standard vtk format, which is especially interesting in case of 3D simulations (see Example 3.8).

If the user specifies, as outlined in Example 3.8, the additional parameter VTK then VTK solution files are written according to the chosen format output option: either “binary_unzip”, “binary_zip” or “ascii”. Note that option “ascii” will lead to large result files and is not recommended in the case of large 3-D industrial applications.

3.2.7 Selection of the outputs

In this section, the types of output files which shall be written by MICRESS® must be specified. These outputs are either binary files which can be viewed with DP_MICRESS (unless another format is specified explicitly in the “Name of output files“ section), or text files which can be opened with standard text editors.

Traditionally, each output type has to be activated or deactivated in an extra line in the driving file and in the requested order by using the corresponding positive or negative keyword, e.g. "out_restart“ or "no_out_restart“ for writing or not writing a restart file (see above). Alternatively, “terse“, “verbose“ or “legacy“ modes can be used. These options will be detailed in the next edition of this manual. In the meantime please consult the MICRESS forum (www.micress.de).

The meaning and the contents of the individual output files in alphabetic order of their extension is documented in chapter 4.3.

3.2.8 Time input data

The times for user-defined intermediate outputs can be specified in this section. For convenience, the user should include an early output time in order to check whether the simulation has started correctly. Series of outputs with a constant interval or factor between the times can be easily defined with 'linear_step' or 'logarithmic_step' (for geometric series). In the example shown here, the user requests outputs for t = 00.25 s and 01.00 s, every 00.50 s until 10.00 s, and every 01.00 s until 25.00 s.

All these input elements can be combined and repeated in arbitrary way. This input section has to be finished using the keyword “end_of_simulation“'.

There are some further options which can be used in this input block for the output times at an arbitrary position:

- “first“: Requests an extra output after the first internal time step.
- “end_at_temperature“: This keyword is followed by a termperature value. It requests an additional output and terminates the execution when the given temperature is reached.

Then, the time step for the phase-field solver has to be defined. It is recommended to use the “automatic“ or “automatic_limited” time step options. In the latter case, minimum and maximum values for the time step as well as optional correction factors (default 0.9) can be further specified which apply to the „phase-field“ and „segregation“ time stepping criteria. Not all of the 4 parameters must be given. Especially, the first value, the minimum time step, can be very useful to improve performance (for more details please use the MICRESS forum at www.micress.de). The parameters for “automatic_limited” can also be read as time-dependent from a file.

For manual definition of the phase-field time step, only the corresponding value in seconds has to be given. As a hint for finding reasonable values, the time-steps for each phase pair according to the automatic criteria is given as screen output during initialization (see Example 3.11).

The section “Time input data“ ends with the definition of the number of iterations for initialisation of the interfaces. During these types of iterations, the sharp interfaces produced by the initial grain structure setting are smoothed by applying the phase-field solver without any chemical driving force. Due to the fact that the stabilisation term, which enforces the interface profile, and the curvature are not separated in

Driving File

```
…
# Time input data
# ============
# Finish input of output times (in seconds) with 'end_of_simulation'
# 'regularly-spaced' outputs can be set with 'linear_step'
# or 'logarithmic_step' and then specifying the increment
# and end value
# ('automatic_outputs' optionally followed by the number
#  of outputs can be used in conjuction with 'linear_from_file')
#  'first'                : additional output for first time-step
#  'end_at_temperature'   : additional output and end of simulation
#                           at given temperature
first
0.1
linear_step 0.5 10.00
linear_step 1.0 25.00
end_at_temperature 1200.
end_of_simulation
…
```

Example 3.9 Time input data: linear_step

Driving File

```
…
# Time-step?
# Options:  fix ...[s]  automatic  automatic_limited
automatic_limited
# Options:  constant    from_file
constant
# Limits: (real) min./s, [max./s], [phase-field factor], [segregation factor]
1.E-7 1.E-3 0.9 0.9
# Number of steps to adjust profiles of initially sharp interfaces [exclude_inactive]?
10
…
```

Example 3.10 Time input data: automatic with 10 iterations for initialisation

Driving File

```
…
Beginning of initialisation
==========
…
Routine init calls routine initTwidth
# tWidth_max( 1 : 1 ) = 0.1187500 s
# tWidth_max( 1 : 2 ) = 0.2375000 s
# tWidth_max( 2 : 2 ) = 0.7916666E-01 s
# tWidth_max( 1 : 2 ) = 11.87500 s
# tWidth_max( 2 : 2 ) = 3.9583333E-04 s
# Maximal value for tWidth = 0.7916666E-01 s for phase-field solver
# Maximal value for tWidth = 9.3571907E-05 s for conc-field solver
…
```

Example 3.11 MICRESS on-screen output for the beginning of initialisation

the phase-field model, curved initial grains will shrink and finally vanish if the number of iterations is too high. For single round particles, an attempt has been made to compensate for that, but in the general case this cannot be avoided. In most cases, there will be no problem, if a small number (~10) of initial iterations is used. On the other hand, the non-conserved phase-field variable gives the opportunity to fine-tune the initial grain fraction during initialisation, e.g. if the "sharp" grain consists of only few grid cells.

3.2.9 Phase data

This section begins with an input of the number of solid phases which will be used in the simulation. Phase number 0 is the “background“ or “matrix“ phase which is implicitly defined and which is assumed to be isotropic (liquid). Then, for each phase specific properties have to be defined. First, the user has to specify whether a stored energy will be defined for this phase and whether recrystallisation will be included into the simulation.

Then, the type of anisotropy for each solid phase has to be specified as “isotropic“, “anisotropic“, „faceted“ or “antifaceted“. If the choice is not “isotropic“, further information on the crystal symmetry is required.

The different growth shapes of crystals are a consequence of their atomic lattice structure (see Figure 3.6) which results in orientation-dependent interface energies and kinetics. The most common type of anisotropy is the cubic metallic anisotropy which in the 2D case is represented by a 4-fold cosine function. Furthermore, there is more specific hexagonal symmetry which may be used e.g. for simulation of Mg or Zn alloys. The “tetragonal“ and the “orthorhombic” symmetries are similar to the cubic symmetry, but with preferred direction.

Driving File

```
...
# Phase data
# =========
# Number of distinct solid phases?
1
# Data for phase 1:
# ------------------------
# Simulation of recrystallisation in phase 1?
# Options: recrystall no_recrystall [verbose|no_verbose]
no_recrystall
# Is phase 1 anisotrop?
# Options: isotropic anisotropic faceted antifaceted
anisotropic
# Crystal symmetry of the phase?
# Options: none cubic hexagonal tetragonal orthorhombic
cubic
# Should grains of phase 1 be reduced to categories?
# Options: categorize no_categorize
no_categorize
...
```

Example 3.12 Phase data: One solid phase with anisotropic cubic crystal symmetry. No grain categorisation is used and grain orientation is defined by 2d angles

If the faceted or antifaceted models are chosen, each facet orientation has to be specified explicitly. This requires the specification of all individual normal vectors of the facets in the local coordinate system. Normal vectors in opposite direction are treated as equivalent.

Furthermore, one has to specify the parameter kappa, which defines the sharpness of the anisotropy function. The antifaceted model is a recently implemented inverse faceted anisotropy function which can be used for needle-like growth.

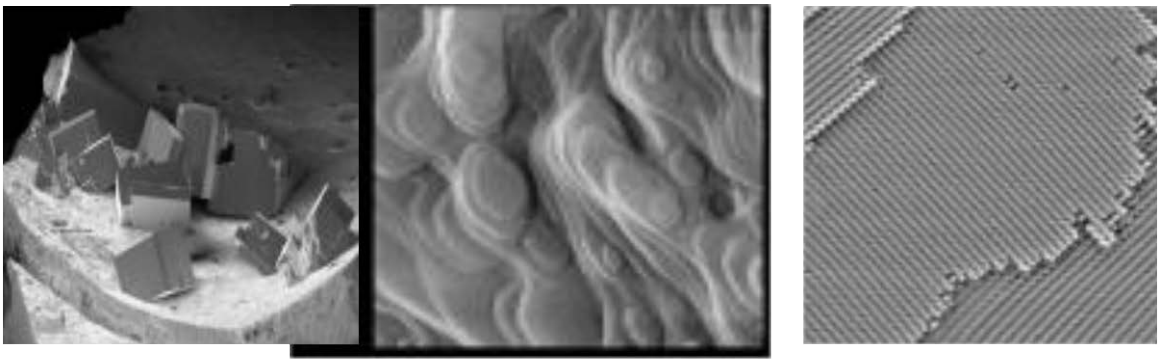


Figure 3.6 Different crystal orientations and facets

Next, the user has to decide whether to use grain categorization. This option allows for sorting the grains of each phase in a user-defined number of orientation "categories". Using categorization, simulations can be sped up, as some operations in MICRESS® are quadratic with respect to the number of grains (or categories). The "categorize" keyword here means that you want to assign grains with identical properties (including orientation) to the same grain number for a given phase. No additional parameter is required after the keyword "categorize".

3.2.9.1 Phase Data for Recrystallisation

For the RX simulation, the user has to select one of the two implemented models:

- *the classical mean RX energy model* based on a main stored energy per grain [J/cm³ or MPa] (see Example 3.13) or its alternative definition via a mean dislocation density ρ per grain [m⁻²] (see Example 3.14)
- *the local RX energy model* based on a dislocation density field, which varies inside each grain (see Example 3.15).

In the case of an alternative input via the mean dislocation density, a direct transformation in a mean RX energy per grain is performed by the program via expression: $E_{RX} = \alpha G b^2 \rho$, where b is the norm of the Burgers vector, G the shear modulus of the considered phase and α the dislocation interaction coefficient.

Driving File

```
…
# Phase data
# =========
# Number of distinct solid phases?
1
# Data for phase 1:
# ------------------------
# Simulation of recrystallisation in phase 1?
# Options:  recrystall   no_recrystall  [verbose | no_verbose]
recrystall verbose
# which recrystallisation model?
# Options: energy  mean_disloc  local_disloc  [all_interfaces]
energy
# energy threshold for recrystallization grain growth [J/cm**3 or MPa]
0.267264
…
```

Example 3.13 Mean energy recrystallization model defined by a mean energy per grain. Grain boundaries of existing grains do not move.

Independent of the adopted RX model, the user has now the possibility to specify via the optional parameter “all interfaces” if the energy difference between existing grains contributes also or not to the RX driving force and, subsequently, if the grain boundaries between old grains can move during recrystallisation or not. Per default, the grain boundary of old grains does not move and do not contribute locally to the driving force. In order to distinguish new grains from old existing ones the user has to specify either an energy [J/cm^3 or MPa] or dislocation density [m^{-2}] threshold.

If the mean RX energy model, defined by a mean dislocation density per grain, is selected, the user must in addition to the dislocation threshold for recrystallization grain growth also provide the Burgers vector [nm] and the shear modulus [MPa] of the considered phase. Note that a dislocation interaction coefficient different from $\alpha = 0.5$ can be specified via an optional parameter.

Note that the output of additional information is activated via the parameter “verbose” behind the “recystall” parameter.

To activate the local RX model the dislocation parameter at the coupling flags (see 3.2.3) has to be specified. In addition to the phase input for the main RX energy variant, defined by a mean dislocation density, the user has now the possibility to take *recovery* before or during recrystallization into account. In fact, a stress recovery coefficient k [MPa] and, optionally, a start time can be specified. Of course, if k = 0, no recovery is taken into account and, per default, recovery starts at the begin of the simulation (t_R = 0 sec). In the case of recovery, the stress of the existing grains decreases according to following empirical law:

$\boldsymbol{\sigma(t) = \sigma_0 - k\, log(t - t_R)}$, where σ_0 is the initial local stress inside each grain. In the program, this law is expressed in terms of a time dependent dislocation density field:

$$\sqrt{\rho} = \sqrt{\rho_0} - \frac{2k}{\alpha G b} log(t - t_R)$$

Driving File

```
...
# Phase data
# =========
# Number of distinct solid phases?
1
# Data for phase 1:
# ------------------------
# Simulation of recrystallisation in phase 1?
# Options:  recrystall   no_recrystall  [verbose | no_verbose]
recrystall verbose
# which recrystallisation model?
# Options: energy  mean_disloc  local_disloc  [all_interfaces]
mean_disloc  all_interfaces
# dislocation threshold for recrystallization grain growth [m^-2]
1.6E+12
# Burgers vector of phase 1? [nm]
#      [dislocation interaction coeff.(default: 0.5)]
2.485    0.5
# Shear modulus of phase 1? [MPa]
54100.0
...
```

Example 3.14 Model variant of the mean energy recrystallization model defined by a mean dislocation density per grain. Grain boundaries of old grains can move.

Driving File

```
...
# Phase data
# =========
# Number of distinct solid phases?
1
# Data for phase 1:
# ------------------------
# Simulation of recrystallisation in phase 1?
# Options:  recrystall   no_recrystall  [verbose|no_verbose]
Recrystall verbose
# which recrystallisation model?
# Options: energy  mean_disloc  local_disloc  [all_interfaces]
local_disloc  all_interfaces
# dislocation threshold for recrystallization grain growth [m^-2]
1.6E+12
# Burgers vector of phase 1? [nm]
#       [dislocation interaction coeff.(default: 0.5)]
2.485    0.5
# Shear modulus of phase 1? [MPa]
54100.0
# stress recovery coefficient of phase 1? [MPa]
#    [ref_time]  [default: 0 sec]
2.0   0.0
...
```

Example 3.15 Local recrystallization model defined by a local dislocation field per grain. Recovery is taken into account and starts immediately.

where ρ_0 is the initial dislocation density field.

3.2.9.2 Grain orientation

At next, the user has to decide in which way grain orientations shall be specified through the rest of the input file. As shown in Example 3.16, following options are now available. one angle in 2D ("angle_2D") and in 3D, a definition via 3 Euler angles ("euler_zxz") or one rotation angle plus a corresponding axis in the 3D space ("angle_axis") or via Miller indices ("miller_indices") or directly defined as a quaternion ("quaternion"). In the program the different grain orientation definitions are transformed in a quaternion. In the case of using the local RX energy model, the user must additionally specify that a *mean* orientation is defined per grain.

Driving File

```
...
# Phase data
# ==========
# Number of distinct solid phases?
1
# Data for phase 1:
# -------------------------
...
# Orientation
# -----------------
# Options: angle_2d  euler_zxz  angle_axis  miller_indices quaternion
euler_zxz
# which kind of grain orientation input?
# Options: mean
mean
...
```

Example 3.16 Grain orientation defined by Euler angles

3.2.10 Grain input

In this section, the microstructure at the beginning of the simulation needs to be specified. The input begins with determining the type of grain positioning. The initial grain structure can either be specified explicitly grain by grain ("deterministic"), by stochastic means ("random") or by reading in a file which represents the initial geometry of the grains ("from_file").

A general rule during grain input is that grain numbers are chosen automatically in a consecutive manner. Grains with a higher number can erase those with lower number if they completely cover them. In case of a partial overlap, the overlapping region by default belongs to the grain with a higher number. Only if the "Voronoi" option is chosen, the overlapping region is distributed between the grains by use of the Voronoi construction.

If a grain radius is defined which is smaller than the grid resolution Δx, a grain consisting of only one interface cell is created which has a grain fraction corresponding to the 3D volume specified by the radius. For those grains, no reasonable curvature can be evaluated using the normal phase-field equation. Therefore, an alternative curvature treatment has to be defined. The "stabilisation" model neglects the curvature as long as the grain is still small. In the "analytical_curvature" model, curvature is calculated from the phase fraction, assuming a spherical morphology. In this case an extra critical radius has to be defined,

Driving File

```
...
# Grain input
# ==========
# Type of grain positioning?
# Options:  deterministic   random   from_file
deterministic
# NB: the origin of coordinate system is the bottom left-hand corner,
#  all points within the simulation domain having positive coordinates.
# Number of grains at the beginning?
1
# Input data for grain number 1:
# Geometry?
# Options:  round    rectangular    elliptic
round
# Center x,z coordinates [micrometers], grain number 1?
0.00000
0.00000
# Grain radius?   [micrometers]
5.0000
...
```

Example 3.17 Grain input: Deterministic grain positioning of one round grain

which determines the maximum value of the curvature for this grain.

If there are no grains to be present at the beginning of the simulation, the user should specify “deterministic“ and define the number of grains at the beginning as 0. In this case, no additional input is necessary in this section.

deterministic:

With this option, first the number of grains at the beginning has to be specified. For each grain, the geometry (round, rectangular or elliptic) and the grain positioning is defined by Cartesian coordinates with the origin at the bottom left-hand corner (in 2D simulations, only the x and z coordinate has to be given). Round grains are defined by their radius, rectangular and elliptic grains by the length along the x and the z-axis. If round grain geometry has been chosen, the curvature model which is to be used in case of small grains has to be specified (see above).

Furthermore, the user has to specify whether in case of overlapping grains the Voronoi construction is to be used and which phase number is associated with the grain. Depending on the properties of this phase, the recrystallisation energy and the orientation of the grain has further to be given.

random

For random grain positioning, an integer for randomization is required as first input. Essentially, this "random seed" assures reproducibility of the initial grain structure when other parameters of the input file are changed.

Driving File

```
…
# Grain input
# =========
# Type of grain positioning?
# Options:  deterministic   random   from_file
random
# Integer for randomization?
122
# Number of different types of grains?
3
# Number of grains of type 1?
2
# Input for grain type 1
# ----------------------------
# Geometry of grain type 1
# Options:  round    rectangular   elliptic
round
# Minimal value of x-coordinates?  [micrometers]
32.
# Maximal value of x-coordinates?  [micrometers]
45.
# Minimal value of z-coordinates?  [micrometers]
20.
# Maximal value of z-coordinates?  [micrometers]
25.
# Minimum grain radius?  [micrometers]
5.00
# Maximum grain radius?  [micrometers]
6.00
…
```

Example 3.18 Grain input: Random positioning of grains

Afterwards, the number of different types of grains has to be specified. By the different types it is possible to e.g. define complex size distributions or to fill different zones of the simulation domain with grains of different size or geometry. For each grain type, the number of grains and the grain geometry must be specified. As in the case of deterministic grain positioning the user can define round, rectangular or elliptic grains. Furthermore, a minimum and a maximum value are required for each-space coordinate in order to define the region over which the grains of this type shall be randomly redistributed. Depending on the type of the geometry chosen, the user has to further specify a minimum and a maximum radius (round geometry) or a minimum and a maximum length size along each-axis (rectangular or elliptic geometry) in order to define the size distribution of the actual grain type.

In the same way as for deterministic input, the curvature model which is to be used in case of small grains has to be specified (if a “round“ geometry has been selected), the user has to specify whether the Voronoi construction is to be used, which phase number is associated with the grain of the actual type and, depending on the phase properties, the recrystallisation energy has further to be given. Additionally, a minimum distance between

the grains (in µm) is required. This parameter, besides the minimum and maximum radius, helps to avoid overlapping of grains. In Voronoi construction, this parameter is helpful to obtain equal size distributions.

from_file

In case of reading the initial grain structure from file, first its name (and path) needs to be specified. For defining the initial grain structure, an image file in ASCII format is required. The geometry of this file has to be given as AnzX and AnzZ (for 2D simulations). These dimensions have not necessarily to be the same as the dimensions of the simulation domain specified at the top of the input file.

The number of grains at the beginning can be either specified explicitly or read from the input file. If the grain properties for the defined number of grains are specified in an extra file, its name and path has to be given.

Otherwise, the grain properties are read in directly from the command line. They can be set

to “identical“, i.e. all grains have the same properties, or they can be read as blocks. The latter would mean grain 1 to 3 and grain 4 to 6 if the number of grains was set to 6.

Driving File

```
...
# Grain input
# =========
# Type of grain positioning?
# Options: deterministic random from_file
from_file
# Filename of initial grain/phase structure
#                                    [VTK_identifier (default=korn)]
Grain_Growth_Microstructure.txt
# Treatment of data?
# (n: none, 1: 1D, f: flip (bottom<->top), t: transpose,
#  or p: 'phase to grains transformation')
none
# AnzX for initial microstructure?
100
# AnzZ for initial microstructure?
100
# Number of grains at the beginning?
# (Set less to 1 then read from input data,
#  with optionally a minimal size, in cells)
2
# Read grain properties from a file?
# Options: input from_file identical blocks
input
# Input data for grain number 1:
# Phase number? (integer)
1
...
```

Example 3.19 Grain input: from file, with no data treatment an 2 grains in the beginning

A 2D or 3D initial grain microstructure, specified in the VTK format, can also be read in as shown in Example 3.20. As different solution types can exist in a VTK file coming from a previous analysis like crystal plasticity FEM, the grain definition must be selected via the optional parameter “VTK_identifier”. Per default, the identifier for the grain structure is “korn” and only if the identifier has an other name, then the optional parameter “VTK_identifier” has to be supplied.

If the local RX model is adopted, the user must then specify the name of the VTK file, from which the local dislocation density field is read. If the identifier “rhoD” is used to characterize the dislocation field, no specific “VTK_identifier” has be introduced. On the other side, if the keyword of the scalar VTK field differs from “rhoD”, then the used keyword name must be provided via the “VTK-identifier”.

Driving File

```
...
# Grain input
# =========
# Type of grain positioning?
# Options: deterministic random from_file
from_file
# Filename of initial grain/phase structure
#       [VTK_identifier (default=korn)]
ReX_korn.vtk   korn
# Treatment of data?
# (n: none, 1: 1D, f: flip (bottom<->top), t: transpose,
#  or p: 'phase to grains transformation')
none
# Number of grains at the beginning?
# (Set less to 1 then read from input data,
#  with optionally a minimal size, in cells)
-1
# Read grain properties from a file?
# Options:  input   from_file  identical  blocks
from_file
#Filename of properties of initial grain structure
Gr_Euler_T950.txt
# Filename of local dislocation densities
#  [VTK_identifier (de-fault=rhoD)
rho_T950.vtk    rhoD
#
...
```

Example 3.20 Grain input: from file, with no data treatment, grain properties (Euler angle) and file name of local dislocation density field.

3.2.11 Data for further nucleation

In praxis, inoculants are often added to technical alloy melts which serve as nucleation agents during solidification. They help to achieve a smaller grain size and to suppress columnar growth. Even if no active inoculation is done, impurities, dislocations or the roughness of the interface structures can lead to nucleation phenomena in all types of technical processes. Unfortunately, apart from the “seed density model“ for heterogeneous nucleation from an inoculated melt, no physical models are available for the complex nucleation conditions in technical alloys. Homogeneous nucleation, on the other hand, will typically occur only at very high undercooling and under extremely clean conditions, like in experiments with levitated drops.

Thus, besides the physically based “seed density model“ for heterogeneous nucleation, MICRESS® provides the user with a pragmatic nucleation model based on a critical undercooling or critical RX energy or critical mean dislocation density (depending on the adopted model variant) which can be further specified with respect to the type of seed positioning, the temperature range, the matrix and substrate phases, the nucleation rate etc. and which allows the user to mimic the complex nucleation circumstances found in technical alloys or processes. All these parameters can be specified in the section “Data for further nucleation“.

The keyword “nucleation“ activates nucleation input and includes an optional parameter in the same line: While “no_verbose“ is the default, “verbose“ triggers additional text outputs on screen which inform the user about the number of potential nucleation events per seed type and how many of those are discarded according to the user-specified parameters like temperature interval or shielding. This option is very helpful for debugging, e.g. for finding out why a given set of nucleation parameters is not leading to any nucleation events during a simulation run. The option “out_nucleation“ in the next line gives the possibility to obtain additional graphical result outputs for the time-step when nuclei are set or a phase disappears. Next, the number of types of seeds must be specified. By using several seed types, different nucleation conditions can be independently defined for different phases on different types of positions, for different temperature intervals etc.

Driving File

```
...
# Data for further nucleation
# ===================
# Enable further nucleation?
# Options: nucleation nucleation_symm no_nucleation
#              [verbose|no_verbose]
nucleation
# Additional output for nucleation?
# Options:    out_nucleation    no_out_nucleation
no_out_nucleation
#
# Number of types of seeds?
4
#
# Input for seed type 1:
# -------------------------------
# Type of 'position' of the seeds?
# Options: bulk region interface triple quadruple [restrictive]
interface
# Phase of new grains (integer) [unresolved |add_to_grain]?
2
# Reference phase (integer) [min. and max. fraction (real)]?
0
# Substrat phase [2nd phase in interface]?
1
...
```

Example 3.21 Further nucleation enabled with one seed type in the bulk

Input for each seed type

For each seed type, a type of positioning must be given. Seeds may be placed in the bulk (=“inner“ part of the grains), in regions, at interfaces, at triple or at quadruple junctions. Unless the additional keyword “restrictive“ is used, the choice of a given keyword includes all keywords which in the options list are found right of this keyword. Thus, “bulk“ or “region“ include all interfaces, triple points and higher junctions, “interface“ includes all triple or quadruple junctions and so on. If “region“ is requested, the coordinate ranges (μm) in micrometer must be given in the following lines.

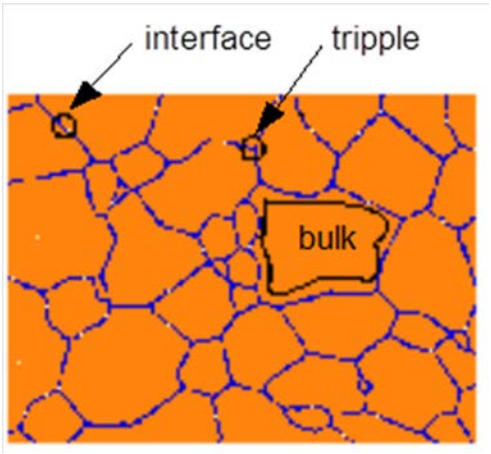


Figure 3.7 The different types of grain positioning: at the interface, at triple junctions and in the bulk.

After that, the phase number of the new grains and the reference phase must be defined. New grains can appear only where the reference phase is present. In case of concentration coupling, the driving force for nucleation is calculated using the local composition in this phase. In solidification simulations e.g. the liquid phase is typically the reference phase.

If the “type of position“ is neither “bulk“ nor “region“, an additional substrate phase (and an optional second substrate phase) is required for further specification of the interfaces where nucleation should occur, and for defining which of the two phases in the interface defines the curvature contribution to the nucleation undercooling (for “interface“ only). This curvature contribution is disregarded if the substrate phase is

identical to the reference phase.

In the same line with the phase number of the new grain two optional models can be specified alternatively, namely “unresolved” or “add_to_grain”. While “unresolved” invokes a model where fine eutectic or eutectoid phases are treated as a continuous phase mixture, “add_to_grain” allows nucleation of a “grain” which already exists in a second place. As the properties of the grain will be overwritten by those specified in the nucleation parameters, this option allows also to change properties of an existing grain. The suboptions “grain_number”, “parent_grain” or “new_set” must be chosen in the next line. For further explanations about these advanced options please refer to chapters 4.7.4 and 4.7.5.

Driving File

```
…
#
# Input for seed type 1:
# ------------------------------
# Type of 'position' of the seeds?
# Options: bulk region interface triple quadruple [restrictive]
interface
# Phase of new grains (integer) [unresolved |add_to_grain]?
2 add_to_grain
# Which grain number the new grains shall be added to?
# Options: grain_number (number) parent_grain new_set
grain_number 0
# Reference phase (integer) [min. and max. fraction (real)]?
0 0.95 1.
# Substrat phase [2nd phase in interface]?
1 3
…
```

Example 3.22 Advanced option “add_to_grain”

Furthermore, the minimal and maximal fraction of the matrix phase where new nuclei are allowed to appear can be optionally specified in the same line with the matrix phase number. This permits for example the user to put new seeds at the outer part of a solid-liquid interface which increases their chances to grow before getting overgrown by the advancing solidification.

In continuation, the nucleation model to be used for further nucleation is defined. MICRESS® uses two nucleation models – the seed density model and the seed undercooling model.

The seed density model is designed for heterogeneous nucleation in a melt, therefore it is only available for “bulk“ or “region“. In all other cases, the seed undercooling model is chosen by default. Both models will be presented separately within the next sections.

Seed density model

The model describes nucleation from the melt, triggered by small seeding particles which may be added intentionally or which may exist as impurities. Essentially, the critical undercooling for nucleation of a given phase on this seeding particle depends on the radius of the seeding particle and the surface energy of the new phase in the liquid. Consequently, if a radius-density distribution of the seeding particles is known, depending on the cooling conditions, the model can predict how many nuclei will be formed.

If the different grains of the new phase grow competitively, like in equiaxed solidification, the latent heat released by the growing particles has to be taken into account. The easiest way to do that in MICRESS® is to specify the global volume heat extraction rate as a temperature boundary condition. Thus, the total amount of latent heat is released globally on the whole simulation domain [1]. A more realistic approach for technical castings is the “Iterative Homoenthalpic Approximation“[2].

At the beginning of the simulation, for all seed types which use the seed density model, discrete positions with discrete radii for the potential nucleation sites are determined and stored. During the simulation run, nucleation is checked only at these predefined places. These seeding particles are not "consumed" by nucleation and cannot move. An exception is the "moving_frame" option in the section "boundary conditions": If this option is selected, the predefined nucleation sites move with the frame, the sites which move out at the bottom of the domain are copied to the top line in order to keep the density of nucleation sites constant.

If "seed_density" is selected, the user first has to specify an integer for randomization. The random number generator has to be initialized with an arbitrary integer number (see Example 3.23). This ensures that, e.g., inserting a new nucleation type would not change the random positions of all other types for a given MICRESS® example.

In the seed density model, the size distribution of potential nucleation sites (seeding particles) is described in terms of classes with different radii [µm] and density [cm^{-3}]. For each class, the number of potential nucleation sites is calculated according to the given density and the volume of the simulation domain.

Driving File

```
...
# Data for further nucleation
# ====================
...
# Which nucleation model shall be used?
# Options:   seed_undercooling   seed_density
seed_density
# Integer for randomization?
111
# How many classes shall be chosen for the critical radius?
6
# Specify radius [micrometers] and seed density [cm**-3] for class  1
1.4  2.E5
# Specify radius [micrometers] and seed density [cm**-3] for class  2
1.2  4.E5
# Specify radius [micrometers] and seed density [cm**-3] for class  3
1.0  1.E6
# Specify radius [micrometers] and seed density [cm**-3] for class  4
0.8  2.4E6
# Specify radius [micrometers] and seed density [cm**-3] for class  5
0.6  5.E6
# Specify radius [micrometers] and seed density [cm**-3] for class  6
0.5  1.0E7
...
```

Example 3.23 Further nucleation using the seed density model

Note: In 2D simulations, the given 3D density is transformed to an equivalent 2D density which has the same average distance between the seeding particles as it would have in 3D.

Using the given density, explicit positions of the potential nucleation sites are determined. The radii (and thus the local critical undercooling for nucleation on each particle) are distributed evenly according to the radius range of each seed class. Inside each class, a random radius distribution is assumed. The radius range corresponds to the radius difference to the next specified class. The finally created numbers and radius ranges for each class can be found in the .log-file.

Note: The seed classes must be specified starting with the highest radius values. At least two classes should be specified, otherwise the automatically assumed radius range may lead to unexpected results.

Optionally, an additional clustering of the randomly distributed seeds can be obtained. This is done by specifying an additional cluster radius and a cluster density in the same line for each class (i.e. four real numbers in each line). The option can be helpful if experimentally a clustering of primary precipitates is observed like in some hypereutectic Al-Si alloys.

Seed undercooling model

In contrast to the seed density model, no explicit potential nucleation sites are predefined and the number of grains which can nucleate during simulation is not limited. Therefore, the user can specify a maximum number of grains which are allowed for each seed type. If this number is exceeded, nucleation of this type stops without

warning (if “verbose“ is not activated, see above).

Next, an explicit radius for the grains of this type has to be specified. If a value higher than the spatial discretisation Δx is chosen, then a grain with the corresponding size and a sharp interface is created.

Note, that with concentration coupling, even if a different phase is nucleated, the new grain adopts the concentration which was present in the concerned grid cells before. Especially with TQ coupling, nucleating „big“ grains most probably will lead to numerical problems and should be avoided!

If a value lower than the spatial discretisation Δx is chosen, a „small“ grain consisting of only one interface cell is created. Usually, a value of 0 will be used to start with the smallest possible fraction of 2 x phMin (see section “Other numerical parameters“). By this way, any kind of concentration imbalance is avoided (given that phMin has been chosen appropriately).

Driving File

```
...
# Data for further nucleation
# ====================
...
# Which nucleation model shall be used?
# Options:   seed_undercooling   seed_density
seed_undercooling
# maximum number of new nuclei 3?
5000
# Grain radius [micrometers]?
0.0
# Choice of growth mode:
# Options:   stabilisation analytical_curvature
stabilisation
# min. undercooling [K] (>0)?
10
...
```

Example 3.24 Further nucleation using the seed undercooling model

Note: The factor 2 given here is the default value of the so-called hysteresis factor. It can be changed manually (see Section 3.2.20).

Under certain circumstances, the user may wish to specify a radius value between 0 and Δx. Then, a grain consisting of a single cell with a fraction of the new phase corresponding to the 3D volume will appear. Afterwards, the small grain model to be used must be specified.

The minimum undercooling specifies at which undercooling nuclei are allowed to form. This undercooling is calculated using the local composition (if concentration coupling is used), temperature and, if applicable, the local curvature of the substrate phase. Depending of the used model variant for RX simulation, the user has to specify then either the critical stored energy [J/cm^3 or MPa] or the critical dislocation density [m^{-2}].

Input for each seed type (continued)

After having chosen the nucleation model to be used in the simulation, the orientation of the new grains has to be specified, if the phase of the new phase has not been set to isotropic (phase input data). A random distribution, a fix value, an orientation range or a parent relation, i.e. a relative orientation to the grain of the local substrate phase, can be chosen. The last option applies only to nucleation at interfaces or junctions. Then, the shield data have to be specified. Shielding means that no further grain of the same phase will be nucleated during the shield time within the shield distance of a previously nucleated grain. No categorisation will be applied to this grain during the shield time (otherwise the shield properties would be lost!). The shield time is also used by the “kill_metastable“ option (see below): No killing is performed during the shield time.

In the same line with the shield time, a shield group number can be specified. Grains with the same shield group number are shielded against each other. The default is the phase number. Note that also phase numbers can be used as shield group number. This allows for establishing shielding between different phases.

After the shield distance an optional parameter for nucleation distance can be entered. This parameter determines the minimal distance between grains of the same phase that nucleate at the same time. If no nucleation distance is given it defaults to the shield distance.

In case of the seed density model, explicit shielding is not compatible with the underlying physical model. The user is requested to specify only a “shield time“ which still is necessary to control the categorisation and „kill_metastable“ functions.

If in section 4.5 “Phase data“ “categorize“ and “anisotropic“ have been chosen for the nuclei phase, the user has to decide whether for this seed type categorisation of the orientation values to orientation categories shall be performed. This is important, if different grains shall be assigned to the same grain number during run-time, because this is only possible if all grain properties including orientations of the grains are identical (see Section 4.5). After the keyword “categorize“, the number of orientation categories can be specified, default is 36 (corresponding to 10° difference between the orientation categories in 2D).

For RX nucleation, the user has to specify also a minimal and maximal energy or alternatively dislocation density of the nuclei. Classically, these energies are zero. In this case, RX starts directly. If these energies differ from zero, RX simulation starts later and an initial microstructure can be built up for example in the case of RX simulation after a solidification analysis. When specifying dislocation densities, their value must be very small and significantly lower than the specified dislocation threshold.

Driving File

```
...
# Data for further nucleation
# ====================
...
# Determination of nuclei orientations?
# Options: random randomZ fix range parent_relation
random
# Shield effect:
# Shield time [shield phase or group number] [s] ?
0.5 105
# Shield distance [micrometers] [ nucleation distance [micrometers] ]?
1.00
# Shall categorization be applied to this seed type?
# Options: categorize {Number} no_categorize
categorize 36
# Nucleation range
# min. nucleation temperature for seed type 1 [K]
0.00
# max. nucleation temperature for seed type 1 [K]
2000.00
# Time between checks for nucleation? [s]
0.01
# Shall random noise be applied?
# Options: nucleation_noise no_nucleation_noise
nucleation_noise
# Factor for random noise?
# (applied as DeltaT -> (1+Factor*(RAND-1/2))*DeltaT)
0.1
...
```

Example 3.25 Nucleation range with random noise to be applied

Further, MICRESS® requests a minimum and a maximum nucleation temperature for the actual seed type. A minimum and maximum temperature should be chosen around the temperature where nucleation is expected. The parameters primarily help to minimize unnecessary nucleation checks and thus to improve performance. In some cases with TQ coupling, checking nucleation too far from the temperature where the new phase gets stable can cause numerical problems. Then, proper values have to be found, depending on the system.

In a first trial, it is wise not to restrict the nucleation range.

The time between checks for nucleation determines the frequency of nucleation checking. If chosen too high, insufficient seeding may occur in spite of high local undercooling. If chosen too small, an unnecessarily high numerical effort and a corresponding performance loss can be the consequence. Note that in case of TQ coupling, each check for nucleation requires a time-consuming call to the TQ interface. A possible performance loss can be monitored using the TabP or TabTQ output.

Noise can be added to the nucleation undercooling. In that case, the coefficient for the noise level and a seed

for the random number generator will be asked.

The noise is applied as $\Delta G \leftarrow (1+k*(random\ -0.5))*\Delta G$ where k is user defined noise amplitude, $0 \leq random \leq 1$ is the value of the random number generator and ΔG is the nucleation driving force.

Input for all seed types

After specification of all seed types, some few inputs remain to be done which apply to all seed types. First of all, if any of the seed types is using random noise, an integer number for randomization has to be given to assure reproducibility. The maximum number of simultaneous nucleations is the number of grains of all seed types allowed to be nucleated in the same time step. A list of possible seeds is created and ordered with respect to the undercooling or driving force for each nucleus. If the maximum number is exceeded, the less favourable seeds are discarded. The option may lead to unexpected results if more than one seed type is defined and therefore should be used carefully. Setting the maximum number of simultaneous nucleations to “automatic“ (=0) removes this check.

In some cases, “stabilised“ small grains can erroneously reach a metastable state at which they stop growing, but also do not vanish because their stabilisation implies a reduced curvature. By enabling the flag “kill_metastable“, the stabilisation of small grains is removed after their shield time has elapsed. This makes sure that “metastable grains“ can vanish correctly. The option “kill_metastable“ is also relevant in case of categorisation, because after „killing“, small stabilized grains are considered as „big“ and such can be assigned to a common grain number.

Important: The “kill_metastable“ flag is only relevant for small grains which use the “stabilisation“ model.

3.2.12 Phase interaction data

In this section the user defines the phase interaction data and the grain boundary properties. Phase interactions can be defined for all pairwise combinations of the phases which have been included in the “Phase data“ section (4.5).

In the standard input sequence, the user is requested to specify all phase interaction data in a fixed order, starting with the phase pair 0/1. Phase interactions which are not used are switched off using the keyword “no_phase_interaction“. No further input is required in this case. The standard input sequence is recommended for less experienced users and for low numbers of phases.

Driving File

```
...
# Phase interaction data
# =================
#
# Data for phase interaction 0 / 1:
# ---------------------------------------------
# Simulation of interaction between phase 0 and 1?
# Options: phase_interaction  no_phase_interaction
#  [standard|particle_pinning[_temperature]|solute_drag]
#   | [redistribution_control]
no_phase_interaction
#
# Data for phase interaction 1 / 1:
# ---------------------------------------------
# Simulation of interaction between phase 1 and 1?
# Options: phase_interaction  no_phase_interaction
#              identical phases nb
#  [standard|particle_pinning[_temperature]|solute_drag]
#   | [redistribution_control]
phase_interaction
...
```

Example 3.26 Example with no phase interaction between phases 0 and 1 and phase interaction between phase 1 and 1

If the number of phases involved in the simulation is high, the input file may get quite long and unreadable, even if most of the theoretically possible phase interactions are switched off. For this case, a “terse“ mode is available, in which only those phase interactions which are needed can be defined in an arbitrary order. The other interactions are automatically switched off. Input is switching to terse mode if the line for the first phase interaction starts with two integers, corresponding to the numbers of the interacting phases. For technical reasons, the section then has to be finished by the keyword “end_phase_interactions“, if the terse mode is used.

Enabling phase interactions means that one of the phases may grow or shrink on the expense of the other. On the other hand, if a phase interaction is switched off, no movement of the corresponding interfaces is possible.

This also concerns the initialisation of the interface: if such an interface is created via the initial grain setting, the interface will stay sharp even if an initialisation is requested (see here). Nevertheless, in case of concentration coupling, there can be diffusion through switched off interfaces. The partition coefficients which are necessary for diffusion through interfaces are accessed from the other phase interactions using a “constant“ approximation. If e.g. interactions are defined for phases 0/1 and 0/2, a simplified description can be derived for the 1/2 interface. This description is stored for each interface cell in the moment of creation (as initial structure or from moving triple junctions) and kept constant during the further simulation steps.

Driving File

```
...
# Phase interaction data
# =================
#
# Data for phase interaction 0 / 1:
# ---------------------------------------------
# Simulation of interaction between phase 0 and 1?
# Options: phase_interaction  no_phase_interaction
#  [standard|particle_pinning[_temperature]|solute_drag]
#   | [redistribution_control]
0 1 phase_interaction
...
end_phase_interactions
...
```

Example 3.27 Phase interaction data with phase interaction in terse mode

Important: If too many phase interactions are switched off, no calculation of the missing partition coefficients may be possible!

Switching off phase interactions can greatly reduce the complexity of a simulation, especially if many phases are included. In solidification, e.g., it is in most cases wise to discard all solid-solid interactions, if a continuation of the simulation (with heat treatment etc.) is not intended. In case of solid state reactions, only the interactions of the precipitations with the matrix and not the interactions between different types of precipitates should be included. If two contacting phases have a common stoichiometric component without solubility range (see here), the interaction between these two phases must be switched off!

When the interaction of one phase with itself is enabled, solid-state interactions between grains of the same phase will be activated. In this case, no chemical driving force is included and the movement will be controlled only by curvature (and by stored energies in case of recrystallisation).

As alternative to “phase_interaction“ and “no_phase_interaction“, the keyword “identical“, followed by the phase numbers corresponding to another interaction (two integers), can be used for abbreviation purposes, if a phase interaction with exactly the same parameters has already been defined previously.

The option “phase_interaction“ can be followed by optional keywords which selects special interaction models:

- “standard“: this is the default interaction
- “particle_pinning“: enables a mesoscale model for particle pinning. If this keyword is selected, an effective mobility model is used which includes the pinning effect of not explicitly resolved particles. As further input, a critical pinning force, which can by defined either by a pinning pressure [J/cm^3 or MPa] (default) or a pinning curvature (μm^{-1}), and a minimal mobility will be requested later in this section.
- “particle_pinning_temperature“: enables the mesoscale particle pining model with temperature dependent input parameters read from an ASCII file.
- “solute_drag“: the solute drag model is based on a two-step hysteresis description of the mobility for the “free“ and “loaded“ interface, as a function of the interface velocity. If this model is selected, a critical transition velocity, a transition range (for smooth transition) and a drag factor has to be specified.
- “redistribution_control“: this keyword activates special models which alter the redistribution behaviour. Currently, models for para-equilibrium, for nple as well as for antitrapping are available. These models are further specified in section 4.10 “Phase diagram – input data“.

In the next line, the driving force options have to be specified. Except for the local RX model, where these options have been specified always, they have been specified only in the case of interaction between different phases. All DeltaG options have to be written in one line, consisting of concatenated pairs of a keyword and the corresponding value (see Example 3.28).

The keyword “avg“ is used to define an averaging of the driving force across the interface. Averaging prevents spreading of the interface if a strong concentration gradient is causing opposite driving forces on both sides of the interface. The user can specify a value between 0 (no averaging) and 1 (maximum averaging).

Averaging of the driving force helps stabilising the interface in case of diffusion controlled transformations, if the resolution is not high enough (i.e., the diffusion length is not big compared to the interface thickness). A typical value is 0.5, the default value is 0.

The keyword “max“ specifies the maximum driving force allowed (value above which the driving force will be cut-off). This value is useful to shrug off some temporary problems during initial transients or to reduce the impact of numerical fluctuations. The value should be chosen high enough in order not to limit kinetics during normal growth.

Important: If a too small value is chosen for the maximum allowed driving force, the movement of the interface can be drastically slowed down!

The “smooth“ keyword has only effect if averaging is specified: The gradient direction along which averaging of the driving force is performed, is randomly rotated with the specified maximum value in degrees. Depending on other circumstances, this option may help to reduce the effect of grid anisotropy on the growth morphology. A typical value is 45, default is 0.

The interface energy, which scales the effect of curvature, can be given either as a constant value (keyword “constant“) or defined as “temperature_dependent“. Temperature dependent values can be read from an ASCII file where the first column represents temperature and the second the interfacial energy. Linear interpolation with temperature is performed on the tabular data. Interface energies have to be specified in J/cm^2.

One of the most important phase interaction parameter is the interface mobility which defines the interface velocity for a given driving force or curvature. The mobility may be defined as constant, temperature dependent or driving force dependent. In the both latter cases, the user will be asked to provide the name of a text file where the kinetic coefficient (second column) is given in tabulated form as a function of temperature or driving force (first column). Otherwise, just a constant value in cm^4/Js is required. In case of some special models (solute drag, particle pinning), additional model parameters have to be included here (see above). As second optional parameter a minimum mobility can be defined which limits reduction of the mobility by automatic time stepping (minimal time step), anisotropy or special models like particle pinning.

Driving File

```
...
# Phase interaction data
# ======================
#
# Data for phase interaction 0 / 1:
# ---------------------------------------------
# Simulation of interaction between phase 0 and 1?
# Options: phase_interaction  no_phase_interaction
#  [standard|particle_pinning[_temperature]|solute_drag]
#   | [redistribution_control]
phase_interaction
# 'DeltaG' options:  default
# avg ... []  max ... [J/cm**3]  smooth ... [degrees]  noise ... [J/cm**3]
avg 0.55 max 100
# Type of surface energy definition between phases LIQUID and 1?
# Options:  constant  temp_dependent
temp_dependent
# File for surface energy coefficient between phases LIQUID and 1?
C:\Desktop\Info.txt
# Type of mobility definition between phases LIQUID and 1?
# Options:  constant  temp_dependent  dg_dependent
constant
# Kinetic coefficient mu between phases LIQUID and 1? [cm**4/(Js)]
...
# Is interaction isotropic?
# Optionen: isotropic  anisotropic [harmonic_expansion]
isotropic
...
```

Example 3.28 Phase interaction with driving force averaging, temperature dependent surface energy, constant mobility of the phase boundary and isotropic phase interaction between liquid and solid

If one of the two phase has been defined as “anisotropic“, the interfacial properties may be defined as “anisotropic”, i.e. as function of the local interface normal. Two alternative model versions can be chosen to describe weak (non-faceted) anisotropy: The default ‘standard’ model offers predefined anisotropy functions. The specific choice of the anisotropy function depends on the crystal symmetry of the anisotropic phase, previously defined by the user. The user can specify anisotropy coefficients $\delta_{\sigma*}$ and δ_μ in the range of $\delta \in [0, 1]$. A typical value for δ_μ in cubic metal systems is 0.05. The more complex ‘harmonic_expansion’ model allows to formulation of anisotropy functions by combination of spherical harmonics in accordance with the selected crystal symmetry. The implemented harmonic functions are listed in Table 4.4.

For combinations of two anisotropic phases, an additional misorientation model is available. If the keyword “misorientation“ is set, the reduction of the mobility and the interfacial energy as function of the misorientation angle or by a given factor can be specified (see Example 3.29). It should be noted that phase 0 (usually liquid) is isotropic by default. An additional optional parameter allows the user to specify the transition angle (in degree) between a low angle grain boundary and a high angle boundary. Per default, this angle is set to 15°. The reduction of the interfacial energy as function of the misorientation angle θ is governed by the new implemented

Read-Shockley law [3]:

$$\boldsymbol{\gamma}(\theta) = \gamma_{HAB}\frac{\theta}{\theta_{HAB}}\left[1 - \ln\left(\frac{\theta}{\theta_{HAB}}\right)\right]$$

where γ_{HAB} and θ_{HAB} are the interfacial energy of high angle grain boundary, specified above, and of the transition angle respectively.

In Fig. 3.8 the interfacial energy reduction, defined by the Read-Shockley model, is outlined. Note that for misorientations below 0.5°, $\gamma(\theta) = 0.1\,\gamma_{HAB}$.

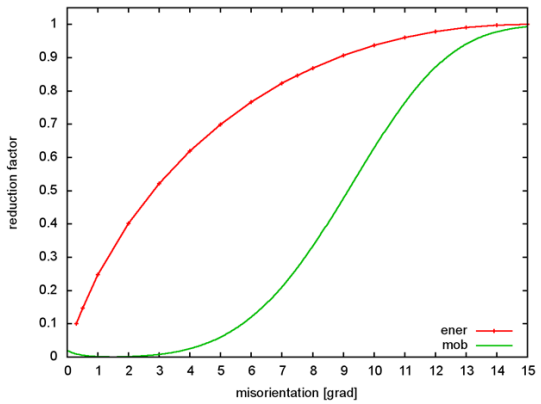


Figure 3.8 Reduction of the interfacial energy and mobility as function of the misorientation angle between two cubic grains.

Driving File

```
...
# Phase interaction data
# ==================
#
# Data for phase interaction 1 / 1:
# ------------------------------------------
# Simulation of interaction between phase 0 and 1?
# Options: phase_interaction  no_phase_interaction
#  [standard|particle_pinning[_temperature]|solute_drag]
#   | [redistribution_control]
phase_interaction
...
#Shall misorientation be considered?
#Options: misorientation  no_misorientation [transition LAB/HAB in
#          degree]
misorientation 15.
# Modification of surface energy for low angle boundaries
# Options: factor Read-Shockley
Read-Shockley
# Modification of the mobility for low angle boundaries
# Options: factor Humphreys
Humphreys
# Parameters B and N of the Humphreys model
5. 4.
...
```

Example 3.29 Reduction of the surface energy and of the mobility of a low angle misorientation between cubic grains, according to the Read-Shockley and Humphreys model respectively.

On the other side, the reduction of the interfacial mobility as function of the misorientation between cubic grains is described by the sigmoidal law suggested by Humphreys [21] (see Fig. 3.8):

$$\mathrm{m}(\boldsymbol{\theta}) = m_{HAB}\left(1 - exp\left[-B\left(\frac{\boldsymbol{\theta}}{\boldsymbol{\theta_{HAB}}}\right)^{n}\right]\right)$$

where m_{HAB} is the interfacial mobility of a high angle grain boundary. B and and n are two model parameters, which can be fitted if experimental data are available: Per default, B=5 and n=4. In order to avoid too small mobilities at very small misorientation angles, the Humphreys law is cutted below 1.5°:

$m(\theta) = 0.02 * exp\left[-C\left(\frac{\theta}{\theta_{HAB}}\right)\right]$ with C = 36.9.

3.2.13 Concentration data

If “concentration coupling“ in the “Flags“ section at the beginning of the input file has been chosen, in this section, the concentration data have to be defined. The concentration data section begins with the number of dissolved constituents.

Afterwards, the user has to specify whether the concentrations units will be atom or weight percent. This definition applies as well to all concentration inputs and outputs as also for the phase diagram data.

Also part of the “Concentration data“ section is the input of diffusion coefficients. Normally, data have to be specified for all contributions, i.e., for each component in each phase (looping automatically through all phases for each consecutive component). In case of many components and many phases, the number of inputs required can be very large, even if for many (e.g. intermetallic) phases no diffusion coefficients are available or need not be specified. Therefore, like in the case of the “phase interaction data“ input, a terse mode is available in which only the positively defined diffusion data are included. The terse mode is automatically invoked if the first input in the diffusion data block is preceded by a phase and component number. In this case, the end of the diffusion data input has to be defined by the keyword “end_diffusion_data“.

For each contribution (phase-component pair), several options for diffusion data input are available. Those data refer to the diffusion flux of the corresponding element in the given phase, which may be caused only by the composition gradient of the same element (diagonal term) or by the gradients of all elements (one line of the diffusion matrix). There are two main keywords (“diagonal” and “multi”) and various specifiers which consist of single characters ("n", "d", "g", "l", "z", "i", "I", or "f") with the following meaning:

- „diagonal“: only the diagonal term is taken into account (Fick’s equation). Correspondingly, only one character of "n", "d", "g", "l", "z", "i", "I", or "f" has to be specified after “diagonal”.

  Example: “diagonal i”

- multi: the flux of the given element is calculated using the gradients of all elements (Fick-Onsager). A string of N characters of "n", "d", "g", "l", "z", or "f" for each dissolved element has to be specified after “multi”

  Example: “multi ggnggg”

- “d“: This keyword indicates that the corresponding term (diagonal or line of the diffusion matrix) will be used and specified directly by the user. If “d“ is selected, the pre-exponential factor and the activation energy (which can be set to 0 for no

Driving File

```
...
# Concentration data
# ===============
# Number of dissolved constituents?   (int)
1
# Type of concentration?
# Options:   atom_percent    (at%)
#            weight_percent  (wt%)
atom_percent
#
# ["Terse Mode": Each line starts with component number and phase number]
# Options:   diagonal [x]   multi [y(1..k)]
#  x: one of the characters "n", "d", "g", "l", "z", "i", "I", or "f"
#  y: chain of "n", "d", "g", "l", "z", or "f" (for each component)
#  default: "g"  resp. "gggg..."
#  Rem: "n":no diffusion, "d": input, "f": T-dep. from file
#       "i":infinite, "I": infinite in each grain
#       from database: "g": global, "l": local, "z" global z-segmented
# Extra option [+b] for grain-boundary diffusion
# Extra line option (prefactor on time step): cushion <0-1>
# Extra line option: infinite_limit [cm**2/s]
# Extra line option: maxfactor_local [real > 1.0] (default: 10.0)
# Finish input of diffusion data with 'end_diffusion_data'.
## How shall diffusion of component 1  in phase 0  be solved?
diff
# Diff.-coefficient:
# Prefactor? (real) [cm**2/s]
2.00000E-04
# Activation energy? (real) [J/mol]
0.0000
# How shall diffusion of component 1  in phase 1  be solved?
database_global
...
```

Example 3.30 Concentration data defined in at% with diffusion

temperature dependence) of the diffusion coefficient have to be specified.

- "n": No diffusion flux will be simulated for the element (and gradient) in the given phase . "diagonal n" is default in the terse input mode.
- "i":Infinite diffusion is assumed for the diagonal term, so that the composition in the phase is mixed completely. Use "infinite" instead of a high explicit value for the diffusion coefficient in order to improve performance (high values of the diffusion coefficients imply very small internal diffusion time steps of the explicit solver, leading to performance loss). Using "i" with "multi" makes no sense and therefore is not allowed. "diagonal i" corresponds to the former "infinite" in the old notation.

  Important: In "i" mode, no distinction between touching and non-touching grains is made. Instead, complete mixing for the specified component between all grains of this phase is performed.
- "I": Complete mixing of the component is performed for each grain individually. Contacting grains interchange rapidly through the common interface region, so that "nearly infinite" diffusion is achieved between all grains of this phase which are in direct or indirect contact. How effective the interchange between contacting grains is depends on the contact geometry and the actual time-stepping. "I" is only allowed with "diagonal" and corresponds to the former "infinite_restricted" (old notation).
- "g": The diffusion coefficients will be calculated by Thermo-CalcTM (via the TQ interface) using a diffusion database. This is only possible if a "GES-file" is available which contains mobility data (see "How to create .ges-files for TC-coupled simulations (overview)", Section 5.8). "g" stands for *global* and means that only one global value for the diffusion coefficient and the effective activation energy will be calculated, based on the actual temperature and the average composition of the element in the given phase. This temperature dependent description will be used for the whole simulation domain. Updates are calculated at a certain interval which is defined by the user at the end of the concentration data input for all contributions.
- "l": Same as "g", but locally defined diffusion coefficients and activation energies are calculated. For performance reasons, these local values cannot be directly calculated from the diffusion database. Instead, the values are interpolated from the local composition dependence with respect to all elements inside the phase. Using "l", diffusion of the diagonal term is described more precisely compared to the "g" option, but higher calculation times are required for retrival of the data from database as well as for interpolation.

  Caution: Using the "l" option can easily lead to demixing diffusion matrixes and divergence of the diffusion solver if extrapolation is going too far! This is especially the case if composition is close to spinoidal like it is typically the case for γ and γ' in Ni-base alloys. Use "maxfactor_local" to control extrapolation limits (see below).
- "z": Same as "g", but instead of one global value for the diffusion coefficient and the effective activation energy a user-specified number of equidistant segments is created along the z-direction of the simulation domain. For each of these segments a global value for the diffusion coefficient and the effective activation energy is calculated. The number of segments has to be specified at the end of the diffusion data input. Use this option in case of strong temperature gradients where large temperature intervals exist inside the domain, or if strong concentration gradients are present in z direction (e.g. diffusion couple).
- "f": read diffusion coefficient T-dependent from file. The file name which contains the diffusion data (first column: temperature, second column: diffusion coefficient) has to be given in an extra line

The described keywords are valid from MICRESS 6.2 onward. The old less systematic keywords ("diff",

“no_diff”, “database_global”, “database_local”, “infinite”, “infinite_restricted”) are still recognized.

A grain boundary diffusion model is available to consider an increased diffusivity of a phase in the interface region to other phases. This increase is expressed as a reduction of the activation energy, leading to a temperature-dependent factor on the diffusion coefficients. The user has to specify this activation energy correction in J/mol as well as the physical interface thickness which is needed to scale to some real thickness. Thus, grain boundary diffusion is only possible for elements in phases which have also diffusion in the bulk and not e.g. for stoichiometric intermetallics without solubility range.

The input works only in the terse mode, i.e. if each diffusion term is defined with a proceeding element and phase number. First, bulk diffusion has to be defined for this contribution. Afterwards, for the same contribution, grain boundary diffusion can be added by using the keyword "+b", followed by a string with length <number of phases +1> consisting of "n" or "b". This string defines for each phase whether in grain boundaries to the actual phase enhanced diffusion is to be considered. For each "b" character, an activation energy difference and a physical interface thickness has to be entered in extra lines

The advanced option **“cushion“** is available to get more control over the internal diffusion time stepping. Normally, an automatic criterion is used which determines the time-stepping per line in the diffusion matrix. The option “cushion“ is only available with “terse“ mode (see above). Adding an extra input line (in arbitrary order with respect to the other diffusion input data) consisting of two integers for the phase and component number, the keyword “cushion“ and a cushion factor (real value) leads to a modification of the diffusion time step for this contribution by the user defined cushion factor. This option is only for advanced users and for debugging purposes. The specified value should be <1 to ensure numerical stability.

The keywords **“infinite_limit”** and **“maxfactor_local”** are also only available in *terse* mode as they specify further information for diffusion terms which must be specified before. “infinite_limit” allows to automatically switch to restricted infinite diffusion (“I”) if the smallest value of the diagonal term in the simulation domain exceeds the value given in the same line after the keyword. “maxfactor_local” allows specifying the maximal factor for extrapolation of diffusion coefficients when using the option “l”. For diagonal terms, this factor applies as well for increasing as for decreasing the local diffusion coefficient while for off-diagonal terms decreasing is allowed down to 0. Default value for this factor is 10. Be careful because for multicomponent diffusion (Fick-Onsager) far extrapolation can easily lead to demixing diffusion matrixes with corresponding divergence of the local compositions!

The option **“factor”** finally allows for a correction of the diffusivity itself by a specified factor. If the diffusivity has been defined before as “multi”, this factor applies to all terms. Thus, it acts as if it were a factor on the mobility of the diffusing species. Please note that the factor also applies to diffusion data specified by “d” and “f”.

If any of the diffusion contributions specified by the user is making use of mobility data included in the .GES5 file (“g”, “l”, “z”), an update interval for these data from the .GES5 file (in seconds) has to be specified. Small intervals typically lead to a more exact simulation of the diffusion processes but increase simulation time, especially if “local“ extrapolation is used. The interval can be either specified as a constant value or be read from a file as time-dependent.

Furthermore, in case of any “z” specifyer having been used, the number of diffusion segments in z-direction has to be given.

3.2.14 Phase diagram – input data

Phase diagram input data have to be defined for all phase pairs for which interactions have been defined in section 4.8 “Phase interaction data“, except for the interactions between identical phases.

In case of uncoupled or temperature coupled simulations (“phase“ or “temperature“ in the “Flags“ section), the pair-wise thermodynamic data are described only by the equilibrium temperature and the entropy of fusion (Example 3.31). For “concentration coupling“ however, thermodynamic data are much more complex. The rest of this chapter deals with this case.

The first type of required information is a list of all the stoichiometric components. Stoichiometric components are those elements in a phase for which no explicit slopes of the phase diagram lines are specified or can be specified. A typical case is an element which has no solubility range in a given phase (line compound), which also implies elements with zero solubility. In such cases, the composition of the element in this phase cannot be used as a thermodynamic condition which is important information if coupling to thermodynamic data is used. This type of stoichiometric condition is automatically detected and needs not to be (but may be) specified by the user.

The user may also decide to define an element with large slope in the phase diagram for numerical reasons as stoichiometric, other reasons could be opposite signs of the solidus and solvus slopes retrieved from the database which also can create numerical problems (pseudo-binary “demixing“).

The definition of the stoichiometric components is available in a compact „terse“ mode: In each line, a phase number and a list of the component numbers which are to be defined as stoichiometric in this phase are specified. The order of specification is arbitrary, different elements of the same phase can also be specified in extra lines. This input block is also used for further special options described below, for the definition of ternary interaction groups, and for setting concentration limits, and is closed by the keyword „no_stoichio“ or „no_more_stoichio“.

Driving File

```
...
# Phase diagram - input data
# ========================
# Equilibrium temperature [K] between 0  and 1
1002.5000
# Entropy of fusion between phase 0  and 1 ?  [J/(cm**3 K)]
1.000
# List of phases and components which are stoichiometric:
#  phase and component(s) numbers
# List of concentration limits (at%):
#  <Limits>, phase number and component number
# List for ternary extrapolation (2 elements + main comp.):
#  <interaction>, component 1, component 2
# Switches: <stoich_enhanced_{on|off}> <solubility_{on|off}>
# End with 'no_more_stoichio' or 'no_stoichio'
no_stoichio
#
# Is a thermodynamic database to be used?
# Options: database   database_verbose   no_database
...
```

Example 3.31 Definition of equilibrium temperature and entropy of fusion

Driving File

```
...
# Phase diagram - input data
# ====================
...
# List of phases and components which are stoichiometric:
#  phase and component(s) numbers
# List of concentration limits (at%):
#  <Limits>, phase number and component number
# List for ternary extrapolation (2 elements + main comp.):
#  <interaction>, component 1, component 2
# Switches: <stoich_enhanced_{on|off}> <solubility_{on|off}>
# End with 'no_more_stoichio' or 'no_stoichio'
solubility_on
stoich_enhanced_on
2 1
Limits 0 1
# Minimal allowed value [at%]
0.
# Maximal allowed value [at%]
20.0
no_more_stoichio
...
```

Example 3.32 Use of expert options for stoichiometric phases

In this section, a few expert options for the use of stoichiometric conditions are available, which are partly not

documented in the standard input file (.in file). They allow a certain “fine tuning“ of the stoichiometric option in case of severe numerical problems with TQ coupling which may occur in complex alloy systems. They should not be used except if absolutely necessary and after contacting MICRESS® experts.

- “Solubility_on“/“solubility_off“: This flag is helpful if components have a finite solubility range, but have to be defined as stoichiometric because of demixing problems., In this case the solubility can be taken into account for the quasi-equilibrium calculation, but not during solute redistribution. If the keyword „solubility_on“ is specified (as extra line), all stoichiometric phases defined after this line are assumed to have a solubility range.
- “stoich_enhanced_on/stoich_enhanced_off“: If a stoichiometric component has a solubility range and has been marked as such (using the keyword „solubility_on“), it is possible to use a more effective extrapolation scheme for redistribution, which assumes a „solidification path“ approximation similar as in the growth restriction theory for nucleation. Although this approximation is not necessarily correct, it can especially during solidification be better than just assuming a constant composition.

The use of stoichiometric conditions, especially when applied to avoid pseudo-binary “demixing“, can easily come to a limit: It is generally not recommended to set a stoichiometric condition for the same element in two phases between which a phase interaction is defined (see Section 3.2.12). With this limitation it may be almost impossible to avoid the effects of pseudo-binary “demixing“ in materials like stainless steels by use of the stoichiometric condition.

Therefore, from MICRESS® version 6.1, a ternary extrapolation scheme has been implemented. Ternary extrapolation gives the user the possibility to account for some of the interactions between the (dissolved) elements which are neglected in case of normal (pseudo-binary) extrapolation and which may lead to a pseudo-binary “demixing“. They are defined in the same input block together with the stoichiometric conditions and can be put in arbitrary order. Using the keyword “interaction“ followed by two component numbers in the same line, a ternary interaction group can be established between two elements and the matrix component. Depending on the total component number, several such ternary interaction groups can be defined in extra lines, but each (dissolved) component can be used only once. The definition of ternary interaction groups applies to all phases and phase interactions.

If using ternary extrapolation for avoiding pseudo-binary “demixing“, ternary interaction groups should include the element for which the “demixing“ occurs, as well as a second element with strong chemical interactions to the first one. If at least one ternary interaction group is defined,

Driving File

```
…
# Phase diagram - input data
# ====================
...
# Is a thermodynamic database to be used?
# Options: database   database_verbose   no_database
no_database
#
# Input of the phase diagram of phase 1 and phase 2:
# ------------------------------------------------------------------------
# Which phase diagram is to be used?
# Options:      linear      linearTQ
linear
# Temperature of reference point? [K]
1149.9000
# Entropy of fusion between phase 1  and 2 ?  [J/(cm**3 K)]
0.14900
# Input of the concentrations at reference points
# Reference point 1: Concentration of component 1 in phase 1 ? [wt%]
0.32000E-02
# Reference point 2: Concentration of component 1 in phase 2 ? [wt%]
0.15811E-03
# Reference point 1: Concentration of component 2 in phase 1 ? [wt%]
0.17399
# Reference point 2: Concentration of component 2 in phase 2 ? [wt%]
0.96288E-01
# Input of the slopes at reference points
...
```

Example 3.33 Phase diagram input data with no coupling to a thermodynamic database

two interaction matrices per phase interaction (one for each direction) are written to the .log file (together with the linearization data, at the time of initialization of the phase interaction) which allow estimating the level of chemical higher order interactions between the dissolved elements. In case of ideal solution, all diagonal matrix elements are 1.0 and the others 0.0.

The option “Limits“ is followed by a phase and component number in the same line, and restricts the allowed composition range of an element in a phase. The minimum and maximum value in at% (always!) is then specified in extra lines. This option is especially helpful if the same phase in the database exists as several composition sets (e.g. FCC_A1 and FCC_A1#2) and “switching“ between the two phases has to be avoided.

After defining the stoichiometric phases, it is time to specify whether Thermo-Calc™ coupling or linearised phase diagrams are to be used. The options are 'database' and 'no_database'. The further input depends strongly on whether a thermodynamic database will be used for the simulation process or not.

If Thermo-Calc™ coupling (via its TQ interface) is NOT to be used, the user has to provide a linear phase diagram description for each phase interaction between different phases defined in the section 4.8 “Phase interaction data“, following the same order. There are two formats which the user can select between, specified by the keywords “linear“ and “linearTQ“. The first format corresponds to the normal linear description which consists of the temperature of the reference point, the entropy of fusion and the reference concentrations and slopes for all elements in both phases.

The matrix component is omitted in this description. The second format is identical to the internal extrapolation format which MICRESS® uses in case of Thermo-Calc™ coupling and thus corresponds exactly to the output format of initial linearisations in the .log output file. In comparison to the normal linearization parameters (keyword “linear“), this format includes additional parameters for the driving force at the reference point and for extrapolation of the temperature dependence of the reference concentrations, and the transformation entropy is replaced by the temperature derivatives of the driving force with fixed composition in phase 1 or phase 2 (dSf+, dSf-).The advantage of the “linearTQ“ input format is that linearization parameter obtained in Thermo-CalcTM coupled simulations (found in the .log or TabLin output file) can be directly used as linear phase diagram description in a subsequent simulation. Another important application is the use for eutectic systems with demixing solid-solid phase interactions: The slopes of the phase diagram lines for the interaction of the two solid eutectic phases typically have opposite signs. This reflects their immiscibility which is enhanced with lowering temperature. This “demixing“ effect can be compensated for by using the “linearTQ“ phase diagram description with different signs for, dSf+ and dSf.

Note: The .log output of the initial linearization parameters for a phase interaction additionally contains information on the transformation enthalpy which is not part of the phase diagram description. For sake of simplicity, during input of the linearization parameters in the linearTQ format, a dummy variable is read for the enthalpy. Thus, the output block in the .log file can be directly transferred to the driving file by “copy and paste“ without removing the transformation enthalpy. If data are taken from other sources (e.g. the .TabLin file), a dummy value has to be included in the corresponding place.

A phase diagram description in the linearTQ format can NOT be exactly translated into a normal linear phase diagram description. If in a “linearTQ“ phase diagram description dSf+ and dSf- have opposite sign, the slopes of ALL elements in the two phases must also have opposite signs.

In case there is a thermodynamic database used in the calculation, the user needs to specify the path to the *.GES5-file (without extension!).

It is possible to increase the size of the TQ-workspace by adding an optional line starting with the keyword "workspace_size" followed by the requested size in kilo-bytes after the specification of the "GES-file". Default is 100 kB. The rest of the input remains unchanged. Increasing the workspace size may be necessary in some few cases which imply many phases and components, especially when the complex γ' phase description of the TTNI- database from ThermoTech is used. Caution: The Thermo-Calc libraries may not always give a meaningful warning if the workspace is too small!

The optional keyword „volume" (also on an own line) enables TQ volume control. This lets the user specify a pressure (other than normal pressure) to be used as initialisation for TQ calculations. The pressure is queried at the beginning of the boundary conditions (see chapter 3.2.19).

As next, the mode for global update of the thermodynamic data has to be set. The options are "manual" "from_file" or "none". In case of "manual" a fixed time interval for updating all phase interactions must be given. Alternatively, the updating intervals can be interpolated from a file as function of time or be switched off ("none").

Then, the thermodynamic description of the pair-wise phase interactions has to be given (in the same order as in the phase_interaction data input): for each interaction, data can be taken from the specified database, from a linearised description ("linear") or according to the "linearTQ" format (see above).

In case of "database", an optional keyword in the same line with "database" can be used to specify a "**global**" or "**local**" phase diagram description: In case of local (default), each interface cell which exists for the given phase interaction bears its own local linearized description obtained and updated from the database according to the specified updating conditions. In the "global" case, only one description is obtained and updated for each interface (i.e. grain interaction) for the given phase pair. The "global" option is less accurate, especially for spatially extended interfaces. In turn, it is computationally much more efficient and reduces numerical noise. The option "**globalF**" automatically detects interconnected areas of the interface and obtains one global set of linearization parameters for each "fragment". The size and number of these fragments can change with time.

Driving File

```
...
# Is a thermodynamic database to be used?
# Options: database  database_verbose  no_database
database
#
# Name of Thermo-Calc *.GES5 file without extension?
GES_Files/AgCuO
# TQ-volume on (enables setting a pressure for TQ)
volume
# Which global relinearisation mode shall be used?
# Options: manual  from_file  none
manual 1.00000E-02
#
# Input of the phase diagram of phase 0 and phase 1:
# ------------------------------------------------------------------------
# Which phase diagram is to be used?
# Options: database [local|global|globalF][start_value_{1|2}] linear
linearTQ
database local
# Relinearisation mode for interface 0 / 1
# Options: automatic  manual  from_file  none
manual 0.001
#
# Input of the phase diagram of phase 0 and phase 2:
# ------------------------------------------------------------------------
# Which phase diagram is to be used?
# Options: database [local|global|globalF][start_value_{1|2}] linear
linearTQ
database globalF start_value_2
# Fraction for initialization of phase 2
init_fraction 0.01
# Relinearisation mode for interface 0 / 2
# Options: automatic  manual  from_file  none
from_file
# File name for the time-dependent relinearisation interval
relinInterval0_1
...
```

Example 3.34 Enabling TQ with volume control and specifying relinearisation option

A further additional keyword (second or third in the row, depending on whether the global/local keyword is used) allows controlling the composition of which phase is used as start value for the iteration of the quasi-equilibrium. The options are "**start_value_1**" and "**start_value_2**", which choose the first or second phase of the phase pair

for delivering the start compositions. Default is the use of the phase with higher fraction if no stoichiometric conditions are set. This option has to be considered as an absolute “expert option“.

As hidden expert option, the keyword “init_fraction” must appear in an extra line following the thermodynamic description of the phase-interaction. This option allows specifying the phase fractions which are applied when initializing the thermodynamic interaction for the given phase pair. This initialization performed either at the beginning of the simulation or when nucleation for a phase which would establish this interaction is checked first time. The initial set of linearization data is written in the .log output. Given the fact that this initialization is crucial for a successfully proceeding simulation, the explicit specification of the phase fractions can be important. Without this option, a fraction of phMin is used for nucleating phases or 0.5 for phases already existing in the initial microstructure. After the keyword “initfraction”, the fraction of the second phase (e.g. phase 2 in the interaction 0/2) can be specified.

Also, for each phase interaction, the relinearisation mode must be defined. A maximum temperature deviation for updating the thermodynamic description can be specified using the option “automatic”. Setting a high value speeds up the calculation by limiting the number of calls to the TQ-library. However, this is achieved at the expense of precision. Alternatively, an updating time interval can be specified for this phase interaction using the keyword “manual” or reading time-dependent from file using “from_file”. This definition of a relinearisation mode per interface can be helpful if one phase interaction, e.g. for a specific intermetallic phase, needs more care.

After having specified all pair-wise inputs related to the use of thermodynamic data, the linking between the phase and component numbers used in MICRESS and in the database has to be made. To do so, the database index of each (MICRESS) component and phase has to be set. MICRESS displays at run-time the list of components and phases available in the selected “GES-file“.

Afterwards the user is prompted for a molar volume of each phase. These values are used for transforming the thermodynamic data per mole to values per volume. Especially in case that latent heat is used, incorrect values can affect the simulation results.

The final input of this section is the temperature for initialisation of the phase interactions between phases present at the beginning of the simulation, i.e. this temperature value is used only for the thermodynamic initialization of those

Driving File

```
...
# Index relations between TC and MICRESS
# ----------------------------------------------------------
# The database contains the following components:
# 1: AL
# 2: CU
# Specify relation between component indices Micress -> TC!
# The main component has in MICRESS the index 0
# Thermo-Calc index of (MICRESS) component 0?
1
# Thermo-Calc index of (MICRESS) component 1?
2
# 0 -> AL
# 1 -> CU
# The database contains 3 phases:
# 1: LIQUID
# 2: ALCU_THETA
# 3: FCC_A1
# Specify relation between phase indices Micress -> TC!
# The matrix phase has in MICRESS the index 0
# Thermo-Calc index of the (MICRESS) phase 0?
1
# Thermo-Calc index of the (MICRESS) phase 1?
3
# Thermo-Calc index of the (MICRESS) phase 2?
2
# 0 -> LIQUID
# 1 -> FCC_A1
# 2 -> ALCU_THETA
# Molar volume of (MICRESS) phase 0 (LIQUID)? [cm**3/mol]
10.000
# Molar volume of (MICRESS) phase 1 (FCC_A1)? [cm**3/mol]
10.000
# Molar volume of (MICRESS) phase 1 (ALCU_THETA)? [cm**3/mol]
10.000
# Temperature at which the initial equilibrium
# will be calculated? [K]
920
...
```

Example 3.35 Further input required in case of coupling to a thermodynamic database

interfaces that already exist in the initial structure.

The result of this initialization process can be seen in the monitoring output (.log-file). Typically, the temperature for initialization is chosen to be identical to the initial temperature at the bottom (see “boundary conditions„ input). In case of trouble with thermodynamic initialization, the user may try slightly different values.

3.2.15 Initial concentrations

There are three options available for the input of the initial concentrations in the simulation domain. The first one is to set explicitly the concentration for each phase by use of the keyword “input“. The second option is to specify only the initial concentration of one phase and to calculate the quasi-equilibrium concentrations for all other phases using the “equilibrium“ keyword. The third one allows reading the initial concentration field from a file, similarly to what is done for the initial grain structure, (keyword “from_file“).

If “input“ is used, the initial concentrations of each constituent in each phase have to be specified, even for phases which might not be present initially. In this case, these values are of less importance and used only under special circumstances in connection with the “start_value_1/2“ option as initial values for iteration. Use this option, if the initial structure shall be out of equilibrium.

Driving File

```
...
# Initial concentrations
# ================
# How shall initial concentrations be set?
# Options:  input  equilibrium  from_file    [phase number]
equilibrium
# Initial concentration of component 1 (C) in phase 0 (LIQUID) ?at%]
1.0000
# Initial concentration of component 2 (MN) in phase 0 (LIQUID) ?at%]
1.0000
# 1D far-field diffusion approximation was set.
# From which distance from the front should the diffusion be
# solved as 1D?  [micrometers]
200.0000
# This distance is equivalent to 200 cells.
# How shall concentration of 1D extension for component 1 be set?
# Options: constant from_file
constant
# How shall concentration of 1D extension for component 2 be set?
# Options: constant from_file
Constant
...
```

Example 3.36 Equilibrium initial concentration with 1D far field diffusion

If “equilibrium“ is chosen, the concentrations only have to be set for one reference phase, the index of which can be specified as optional parameter in the same line (default is phase 0. Note that, strictly speaking, this condition is a quasi-equilibrium, i.e. there still might be a driving force (parallel tangent construction in case of TQ coupling).

In case “1d far-field diffusion approximation“ was set in section “Flags and Settings“, some additional input is necessary (marked in red). This input includes the distance from the front in µm (defined by the highest position in z direction of any grain of a phase different from 0) from which diffusion shall be solved in 1D and the initial concentration definition of the 1D extension.

Each component can be defined as “constant“ or “from_file“. In the latter case, the initial concentration profile of the 1D extension is interpolated from any ASCII file containing one-dimensional information of distances (in µm) and corresponding concentration values. If the keyword “from_file“ is chosen, the name and path of the file need to be specified. The file e.g. could have been written from earlier MICRESS results in ASCII format using DP_MICRESS. In the same line as the “from_file“ keyword, a phase number should be given (default is phase 0). In case of TQ coupling, a composition has to be specified for this phase, serving for initialisation of the equilibria at the beginning.

NOTE: This concentration value has no influence on the initial concentration distribution! Afterwards, like in the case of reading initial grain structures from file, a character must be given which specifies the type of data treatment (n: no action, 1: 1D data, f: mirror top to bottom, t: transpose). Then, the size of the input file has to be given. In case that the geometry of the input file does not correspond to the size of the simulation domain, the data are automatically interpolated.

NOTE: In case of reading in experimental concentration data, numerical problems may occur with TQ-coupling if the data are noisy or not fully consistent with the initial grain structure!

Driving File

```
...
# Initial concentrations
# ================
# How shall initial concentrations be set?
# Options:  input  equilibrium  from_file     [phase number]
from_file 1
# Concentration of component 1 (C) in phase 1 (FCC_A1)? [at%] (for
initialisation of TQ)
2.00
# File for initialisation of concentration profile of component 1?
Example_conc1.txt
# Treatment of data?
#(n:none, 1:1D, f:flip (bottom<->top), or t:transpose)?
n
# AnzX for initial concentration field of component 1 ?
500
# AnzZ for initial concentration field of component 1 ?
400
# Concentration of component 2 (CR) in phase 1 (FCC_A1)? [at%] (for
initialisation of TQ)
0.500
# File for initialisation of concentration profile of component 2?
Example_conc2.txt
# Treatment of data?
#(n:none, 1:1D, f:flip (bottom<->top), or t:transpose)?
n
# AnzX for initial concentration field of component 2 ?
500
# AnzZ for initial concentration field of component 2 ?
400
...
```

Example 3.37 Reading initial concentrations from file

3.2.16 Material data for fluid flow

If [flow] or [flow_coarse] was chosen earlier as a coupling flag the user will be asked for material data concerning fluid simulation in this section. For an overview and a more detailed description of the flow solver see section 4.15.

First entered is the kinematic viscosity ν, it is connected to the dynamic viscosity μ by the equation $\nu=\mu/\rho$. The density ρ, entered next, determines the relation of inertial forces to frictional and buoyancy forces.

To model concentration driven convection the partial density change with composition, $d\rho/dc$, has to be entered for each component. An optional reference concentration can be entered in the same line. If any density changes are different from zero strength and direction of “gravity” is to be entered.

For the 3D case the x, y, and z components of the gravity vector should be entered, while for the 2D case only the x and the z component are entered. Note that positive z-components make the gravity vector point to the top of the simulation domain.

The final h^* parameter determines the strength of friction in the phase-field region. If “auto” is chosen an optimal value is determined automatically for the given interface thickness. In most cases a value of 21.0 is a good choice.

Driving File

```
...
# Input of material data for flow calculations
# ========================================
#
#
# Phase          0
# ---------------------------------------------------------
# Viscosity [cm**2/s], phase           0
1.8910000000000000E-002
# mean density [g/cm**3], Phase           0
3.70000000000000
# partial density change with composition [g/cm**3/%]
# dRho / dC, component           1
-1.77D-2 45D0
# Gravitational vector [cm/s**2] (real) (real) 3D:(real)
0.0 13000
# Friction Parameter h*: (real) | auto
21.0000000000000
#
...
```

Example 3.38 Input of material data for fluid flow

3.2.17 Material data for elastic calculation

The following section applies only for “stress“ and “stress_coupled“ simulations. The first input here is the phase number of the matrix phase. The strain and the calculation of its eigenvalue will be referenced to this phase.

For each cubic or isotropic phase its molar volume and the three elastic coefficients $C_{1!}$, C_{12}, C_{44} of the Hooke matrix are necessary to be specified. In the case of a thermo-elastic analysis also thermal expansion coefficients in the X, Y, Z directions have to be specified for the considered phase.

Driving File

```
...
# Input of material data for elasticity calculations
# ==================================
# Phase number of the matrix phase (int)
1
# Notation for Eigenstrain?
# Options: volume  matrix
volume
# Abstraction of the 3D continuum
#   Options:  plane_stress  plane_strain
plane_stress
#
# phase 0
# ----------
# molar volume [cm**3], phase  0
4.0000E-008
# Thermal expansion coefficients
# along the principal axes xx [1/K], yy [1/K], zz [1/K]?
#    (real)  (real)  (real)
0.  0.  0.
# Notation for elastic coefficients?
# Optiones: c_ij  Voigt materialParameter
c_ij
# Elastic components (Voigt-notation)
# C11 [MPa] ( >0 )      , phase  0
23300.0000
# C12 [MPa] ( >0, <C11 ), phase  0
13300.0000
# C44 [MPa] ( >0 )      , phase  0
11600.0000
...
```

Example 3.39 Input of elastic material data

3.2.18 Parameters for latent heat and 1D temperature field

Latent heat describes the amount of energy released or absorbed during phase transition. In concentration coupled simulations, the latent heat option is typically applied to equiaxed growth with negligible temperature gradients. In this case, the latent heat is released averaged over the calculation domain (DTA approximation). MICRESS® also allows the use of latent heat in connection with thermal gradients, but in such cases, rather the use of the 1d_temp option (3.2.3) is recommended.

At the top of this input section, the user has to choose between the keywords “no_lat_heat“, “lat_heat“ and “lat_heat_3d“.

If latent heat is used, the enthalpy of each phase (including the phase 0), as well as its specific thermal capacity need to be specified. In case of TQ-coupling, the values are read automatically from the database. In case the 1d_temp option has been selected, heat conductivity input is necessary for each phase. Like in other places of the input file, one can define a constant value or input a temperature dependent heat conductivity which is read from a text file (see e.g. section 3.2.12 Phase interaction data).

Driving File

```
...
# Parameters for latent heat and 1D temperature field
# ====================================
# Simulate release of latent heat?
# Options:      lat_heat     lat_heat_3d [matrix phase]
lat_heat_3d 0
# Type of thermal conductivity definition for phase  0 (LIQUID) ?
# Options:    constant     temp_dependent
constant
# Thermal conductivity of phase  0 (LIQUID) ?  [W/cm/K]
1.3000
# Type of thermal conductivity definition for phase  1 (FCC_A1) ?
# Options:    constant     temp_dependent
constant
# Thermal conductivity of phase  1 (FCC_A1) ?  [W/cm/K]
1.2000
# Simulation with release of pseudo-3D latent heat of phase 1
#     (FCC_A1)?
# Options:      pseudo_3d [crit. matrix fraction]     no_pseudo_3d
pseudo_3D 0.75
...
```

Example 3.40 Definition of 3D latent heat with constant thermal conductivity

The keyword “lat_heat_3d“ switches on a correction of the amount of latent heat which may be wrongly predicted in some types of 2D simulations (see section 4.11). An additional phase number input allows specifying a reference phase for this correction, otherwise phase 0 is assumed. If “lat_heat_3d“ is chosen, the user can activate or deactivate the 2D/3D correction for each further phase by the keywords “pseudo_3D“ / “no_pseudo_3D“ followed by the critical fraction of the matrix phase for limiting the correction (Section 4.11). The use of latent heat as well as the “1d_temp“ option has further implications for the definition of the boundary conditions (see below).

3.2.19 Boundary conditions

If a thermodynamic database is used and TQ volume control is enabled (see section 3.2.14) the pressure condition for TQ calculations is asked at the start of the boundary conditions section.

Two classes of boundary conditions are distinguished in MICRESS®:

Thermal boundary conditions

Thermal BCs have to be specified if temperature is not explicitly simulated.

- temperature vs. time
- net heat flow vs. time
- temperature gradient vs. time

Driving File

```
...
# Boundary conditions
# ====================
# Pressure for TQ-calculations (Pa)?
21230.0
# Type of temperature trend?
...
```

Example 3.41 Pressure condition for TQ

- 1d-temperature field

Conditions for boundaries of the calculation domain

- for the phase-field parameter
- for the concentration field
- for the 1d-temperature field, the stress field, flow field etc

This section starts with the specification of the temperature boundary condition, if neither “temperature“ coupling nor “1d_temp“ is chosen in section “Flags and Settings“ at the top of the input file. Three alternatives are available to describe the type of temperature trend: If the flag “linear“ is chosen, the number of connecting points has to be defined. Then, first the initial temperature at the bottom of the simulation domain and the initial temperature gradient in z-direction are requested. Afterwards, for each connecting point, the time the temperature at the bottom and the temperature gradient in z-direction have to be specified. The temperature and temperature gradient will be interpolated linearly between these transition points. In this way, an arbitrarily complex temperature-time profile can be applied.

Driving File

```
...
# Boundary conditions
# =================
# Type of temperature trend?
# Options:  linear   linear_from_file   profiles_from_file
linear
# Number of connecting points?   (integer)
2
# Initial temperature at the bottom?  (real)  [K]
1150.0
# Temperature gradient in z-direction?  [K/cm]
0.0
# 1 Connecting point, time  t [sec]?
2.5
# 1 Connecting point, temperature
1125.
# 1 Connecting point, temp. gradient in z-direction [K/cm]
5.0
# 2 Connecting point, time t [sec]?
7.5
# 1 Connecting point, temperature
1000.0
# 2 Connecting point, temp. gradient in z-direction [K/cm]
25.0
...
```

Example 3.42 Definition of linear temperature trend and connecting points

If the number of connection points is set to 0, only the initial temperature at the bottom, the temperature gradient in z direction and a constant cooling rate in K/s are requested.

Using the flag “linear_from_file“, it is also possible to read the same information from a file consisting of three columns: the time in seconds, the temperature at the bottom in Kelvin, and the gradient in Kelvin per centimetre. This allows a more compact input for complex temperature-time input data.

While the first two options always lead to linear temperature profiles in z-direction, the option “profiles_from_file“ allows to read time-dependent non-linear temperature profiles from a series of ASCII data files (or from different columns of one data input file). Again, the number of connection points has to be specified, and for each of those points, the time and a file name, optionally with specification of the used

Driving File

```
...
# Boundary conditions
# ===============
...
# Moving-frame system in z-direction?
# Options:    moving_frame    no_moving_frame
moving_frame
# Should the distance or the bottom temperature be
# used as criterion for moving frame?
# Options:  distance [matrix phase (negative: special phase)]
#           temperature
distance
# At which distance from the upper boundary should the frame
# be moved?  (real)  [micrometers]
200.0000
#
# Store data shifted out of moving-frame system?
# Options:    out_moving_frame    no_out_moving_frame
no_out_moving_frame
...
```

Example 3.43 Definition of a moving frame with the distance criterion and no data storage

columns, has to be given. The example file “Grain_Growth_Profiles.dri“ illustrates the use of this option for simulation of grain growth in the heat-affected zone of a welding (see “MICRESS® User Guide Volume IV: MICRESS Examples“).

If the release of latent heat is enabled and “1d_temp“ is not selected, specification of the thermal boundary condition is made via the input of a heat flux (in Js/cm³) instead of a cooling rate or temperature trend (“DTA approximation“). Like in the case of the temperature trend, an input of complex heat flux-time relations is possible via connection points or by reading from a file. Of course, the option “profiles_from_file“ is not available in case of using latent heat.

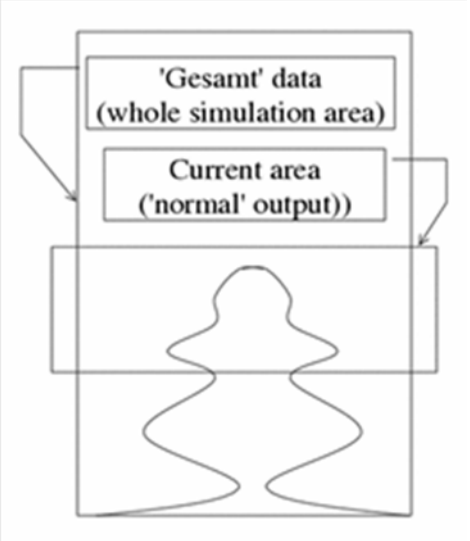


Figure 3.9 The moving frame: the current area within the whole simulation area

As next, the “moving frame“ options have to be set. This feature is very useful for simulations related to directional solidification, as it allows the reduction to a smaller simulation domain which follows the movement of the solidification front. This is shown schematically in Figure 3.9 and in the example of the directional solidification of a conventional Ni-base superalloy in Figure 3.10. This option is not available for the use of latent heat without 1d temp because it is not compatible with the assumptions of the “DTA approximation“. If the flag “moving frame“ is selected, the user has to select the criterion which controls the movement of the simulation domain. Available criteria are “temperature“ or “distance“. In case of “temperature“, a critical temperature is requested. If the temperature at the bottom falls below this value, the domain is moved in z-direction until the bottom temperature reaches the critical value. In case of a constant cooling rate, this option leads to a constant moving velocity like in a typical Bridgman furnace experiment.

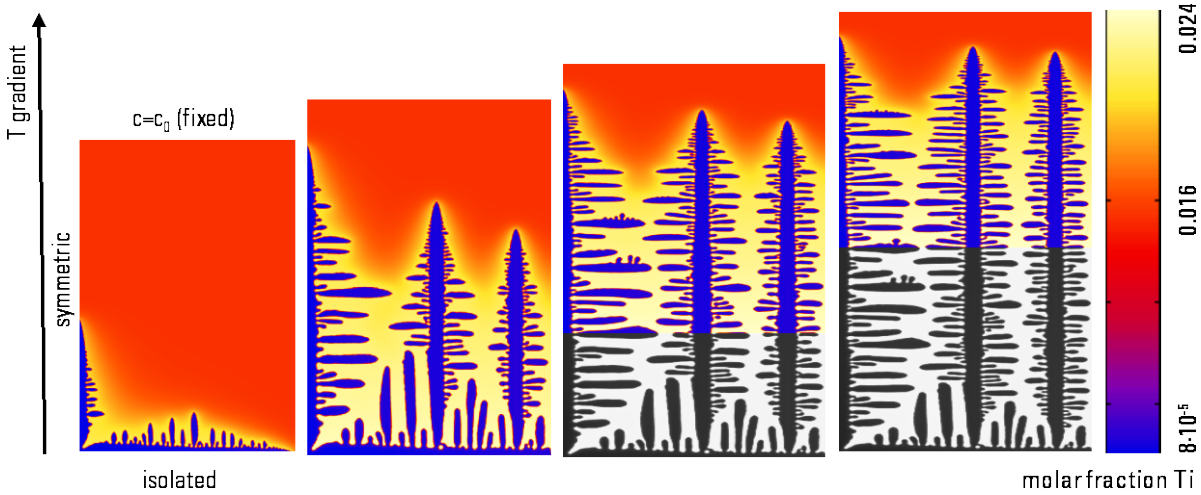


Figure 3.10 2D simulation with moving frame: directional solidification of IN706

If “distance“ is selected, the distance of the solidification front (more exactly the highest position of any object with phase number >0) from the top of the simulation domain is used as criterion for movement of the simulation domain. Thus, for example, a constant distance of a dendrite tip from the top can be achieved. In both cases, the corresponding value has to be given in the next line after the selected keyword. If the “out_moving_frame“ flag is set, a “Gesamt“ (global) data file is created additionally for all selected graphical outputs.

The corresponding files have a “G“ appended to their “root name“ (e.g.: “SimulationG.phas“ under Linux). They consist of the simulation domain for the latest output (the coloured part in Figure 3.9. to which all the information of the grid cells that have moved out from the domain during simulation is appended (the grey part in Figure 3.10). Consequently, they can be much larger than the normal outputs and have their own geometry file (“SimulationG.geoF“).

NOTE: using the “moving frame“ option is very reasonable in combination with the “1D far field“ option. This combination allows the simulation of stationary processes with long-range diffusion profiles.

If the option “1d_temp“ has been selected at the beginning of the input file (see Section 3.2.3), a one-dimensional temperature field is applied as “extended“ temperature boundary condition. The initial temperature field can either be read from a text file or be specified in linear approximation by input of a value for the upper and lower end of the 1d-temperature field. The 1d-temperature field must be positioned such that the 2D/3D simulation domain is located completely inside this field. The user has to specify the relative position of the bottom of the 1d-temperature field relative to the bottom of the microstructure domain.

Important: the initial position of the bottom of the 1d temperature field is always negative with respect to the micro-domain.

Driving File

```
...
# Boundary conditions
# ===============
...
# Boundary conditions for phase field in each direction
# Options: i (insulation) s (symmetric) p (periodic/wrap-around)
#          g (gradient)   f (fixed)     w (wetting)
# Sequence: W E (S N, if 3D) B T borders
ppii
#
# Boundary conditions for concentration field in each direction
# Options: i (insulation) s (symmetric) p (periodic/wrap-around) g
(gradient) f (fixed)
# Sequence: W E (S N, if 3D) B T borders
ppii
#
# Boundary conditions for 1D temperature field bottom and top
# Options: i (insulation) s (symmetric) p (periodic/wrap-around) g
(global grad) f (fixed) j (flux)
# Sequence: B T
fi
# How shall temperature in B-direction be read?
# # Options:   constant    from_file
constant
# Fixed value for temperature [K]
298.00
# Fixed value for heat transfer coefficient [W/cm2K]
1.5
# Please specify for the 1D temperature field, which enthalpy
# below the calculation domain should be present!
...
# Please specify for the 1D temperature field, which value of Cp
...
# Please specify for the 1D temperature field, which
#   value of the heat conductivity
...
# Please specify for the 1D temperature field, which enthalpy
# above the calculation domain should be present!
...
```

Example 3.44 Defining the boundary conditions

The boundary conditions of the simulation domain have to be set for the phase-field variables, the concentration, temperature, fluid flow and displacement fields depending on the type of coupling which has been defined at the beginning. The MICRESS® boundary conditions are defined by a text string with length 4 or 6 which represent a sequence of key characters. The characters specify the type of boundary condition, their sequential order addresses the different sides of the simulation domain (west-east-bottom-top for 2D and west-east-south-

north-bottom-top for 3D). The following conditions are available (see Figure 3.11):

insulation ("i"): The boundary cell (the first cell outside of the simulation domain) is assumed to have the same field value (e.g. phase-field variable) as its direct neighbour (the outermost cell of the domain). The name of the flag reflects the fact that no gradients and, thus, no fluxes exist between the boundary cell and its neighbour inside the simulation domain.

symmetric ("s"): defines the field value of the boundary cell to be identical to its second neighbour in the simulation domain, thus implying a symmetry plane through the centre of the outermost cells of the domain. This condition is similar to an isolation condition which is shifted by half a cell.

periodic ("p"): with this condition, the field value of the boundary cell is set to the value of the outermost cell on the opposite side of the simulation domain. Thus, objects like dendrites which touch one side are continued on the other side. The periodic condition preserves the field balance.

gradient (g): the field value of the boundary cell is extrapolated from the first and second neighbour inside the domain. The use of this boundary condition is allowed for all fields (concentration, temperature, phase-field) but not always reasonable. The gradient condition for phase-field is very useful for grain growth. If "periodic" is not suitable for any reason – the flag "g" should be the best choice for minimising the impact of the boundary condition on the grain structure. Be prepared to get strange effects while using "g" with the concentration field, if phase boundaries are touching the domain boundary!

fixed ("f"): Uses a fixed value for the boundary cell. This value is requested in an extra input line. Naturally, the "f" condition does not preserve the average of the field value. A typical application of the fixed condition for the concentration field is directional solidification with moving frame (fixed condition for top boundary).

Please note that in case of using a 1D extension of the concentration field (by selecting the "1d_far_field" option in the "Flags" section), the top boundary condition for the concentration field is automatically shifted to the top of the 1D extension!

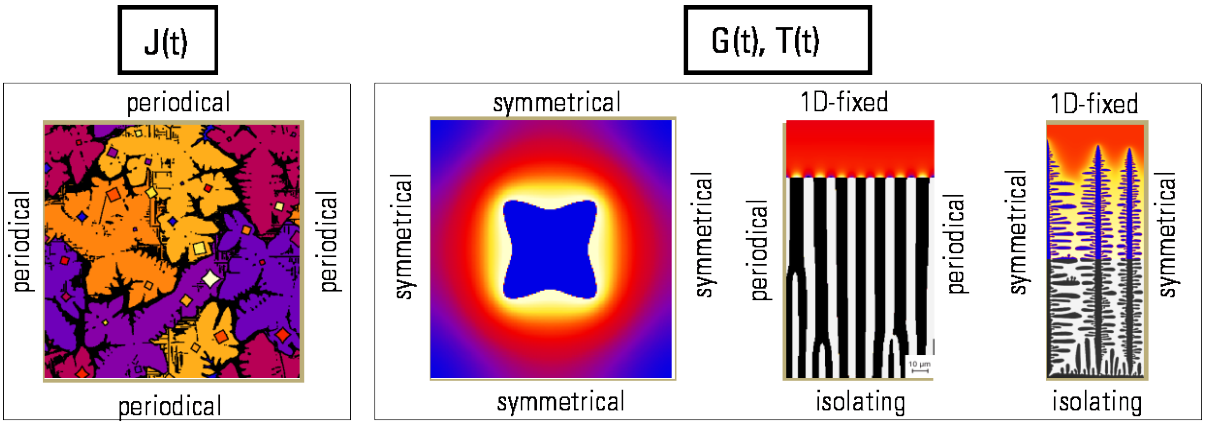


Figure 3.11 Graphical visualisation of the boundary conditions

If the option “1d_temp“ has been selected at the beginning of the input file (“Flags and Settings“), at this place the boundary conditions for the 1D temperature field have to be specified. The user can select between insulation (i), symmetric (s), periodic (p), global gradient (g), fixed (f) and flux (j). While “i“, “s“ and “p“ have already been explained above, the other conditions are either new or have further implications or a slightly different meaning:

global gradient (“g“): This condition establishes a given global temperature gradient between the actual boundary and the opposite boundary. This modified gradient condition is especially useful for coupling to external process simulation results: If temperature vs. time and the thermal gradient are known from a macroscopic process simulation (or a corresponding experiment), a time-dependent fixed “f“ condition (see below) can be applied on one side of the 1D temperature field, and the “g“ condition on the other side to maintain the gradient. The definition of “g“ on both sides is not allowed!

fixed (“f“): The definition of this condition corresponds to that of the fixed condition for the normal simulation domain. But, not only of a fixed temperature value, but also a temperature-time profile can be read from a text file using the “from_file“ option. If a constant temperature is chosen, a heat transfer coefficient is requested additionally, allowing the definition of a heat transfer condition to an external medium with fixed temperature. If the value of the coefficient is <>=0, the temperature value is used as fixed condition instead.

flux (“j“): This condition allows the assumption of a constant or time-dependent flux [W/cm^2] as boundary condition.

After defining the boundary conditions of the 1D-temperature field, the user has to choose whether to apply constant thermo-physical data (enthalpy, heat capacity and thermal diffusivity) for this temperature field, or to specify files where these thermophysical data shall be read from as a function of temperature. All three thermodynamic quantities are read separately for the part of the 1D-temperature field which lies above and below the micro-simulation domain. This can be important if e.g. a strongly undercooled columnar dendritic front is simulated.

Driving File

```
...
# Boundary conditions
# ===============
...
# Unit-cell model symmetric with respect to the x/y diagonal plane?
# Options:    unit_cell_symm   no_unit_cell_symm
no_unit_cell_symm
...
```

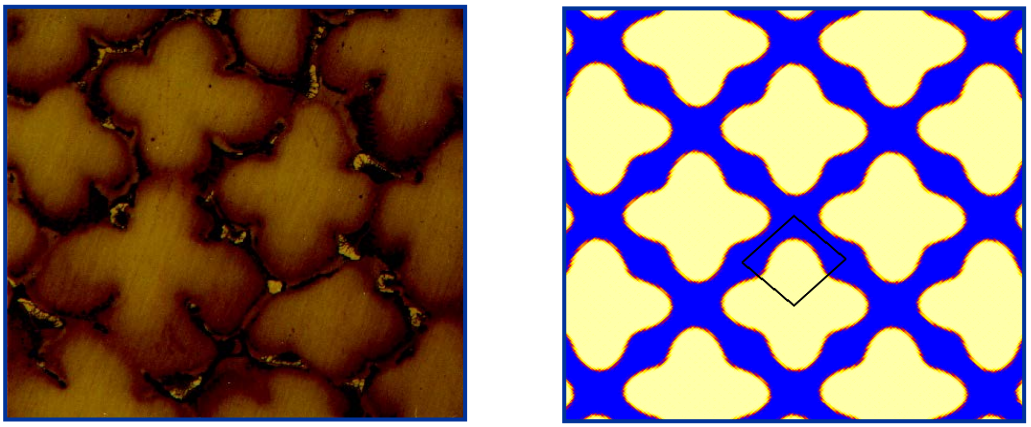
Example 3.45 Definition of no unit cell symmetry

In many cases, all this thermophysical information can be taken (iteratively) from the dTLat output of a prior MICRESS simulation [2]. Alternatively, the thermo-physical data file may be created using Thermo-Calc™. Some additional information on the parameters for latent heat and 1D temperature field is available in section 4.12.3.

In case of concentration coupling, the user is asked to define the symmetry of the unit cell (Example 3.45). With the “unit_cell_symm“ flag, one can try to keep some symmetry with respect to the x/y diagonal (see Figure 3.12). The “no_unit_cell_symm“ flag disables this option.

If “unit_cell_symm“ is used in case of a quadratic or cubic simulation domain, a four-fold symmetry will also be maintained by respective symmetric nucleation.In case of stress coupling, additional boundary conditions are required. The available options are “constant_volume“, i.e. zero displacement of all boundaries, “free_expansion“, i.e. bottom left corner fixed with free expansion in all directions and “parallel_expansion“, i.e. bottom left corner fixed, free expansion along main axes.Furthermore, external stress and strain acting on the simulation domain can be defined. The boundary conditions are “fix_isostatic_pressure“, “fix_isostatic_strain“,

"fix_normal_pressure" and "fix_normal_strain". For the "isostatic" cases, either the normal pressure at the domain boundaries or the normal displacement are fixed. The values should be specified in a second line (real numbers, units are [MPa] or [%], respectively). The "normal" variants allow the definition of normal displacement or normal stress separately in each direction.



real structure (CMSX-) idealised struture

Figure 3.12 Isothermal unit cell model for a 2D cross-section through the mushy zone

If the flow module is activated the user is asked to provide two sets of boundary conditions, one for flow velocity and one for pressure. The available options are listed in Table 3.1. The 3rd column of the table for pressure boundary conditions lists the velocity boundary conditions that should preferably be combined with a specific pressure boundary condition. If periodic boundary conditions are chosen they should be applied to the opposite boundary as well and to velocity as well as pressure. In this case the user is asked to provide a pressure difference for the boundary pair.

For the "fixed", "symmetric", "iso" and "parallel" velocity conditions "no condition" can be chosen for pressure with a "-" (dash). The "gradient" boundary conditions use a linear extrapolation of velocity or pressure on the boundary, to provide a fixed pressure gradient use the "neumann" boundary condition. Please see section 4.15 for more details on boundary conditions.

Table 3.1: Velocity and pressure boundary conditions for fluid dynamics.

Velocity boundary conditions	
Letter	Meaning
p	periodic, same velocity as opposite boundary
f	fixed velocity, provide components in μm/s
s	symmetry plane through cell center
i	iso, symmetry plane at the boundary
l,=	parallel, frictionless boundary condition
o,+	orthogonal flow to the boundary
e	free in/out flow
g	gradient extrapolation of the velocity

Pressure boundary conditions		
Letter	Meaning	velocity bc
p	periodic with fixed offset	p
f	fixed	o, e, g
s	symmetric	o, e, g, s
i	isolated	o, e, g, i, l
n	neumann, fixed gradient	o, e, g
g	gradient extrapolation	o, e, g, f
-	no boundary condition	f, s, i, l

3.2.20 Other numerical parameters

In this last section, some purely numerical parameters for the phase-field, concentration, flow and stress solver are defined. The maximum number of concentration solving per phase-field iteration has only to be defined, if concentration coupling is used in combination with a fixed phase-field time step. An error message is given if the number is too small to be consistent with the time-step criterion of the explicit diffusion solver. This input is not required when automatic time-stepping is used.

Driving File

```
...
# Other numerical parameters
# ====================
# Phase minimum?
1.00E-04
# Interface thickness (in cells)?
7.5
...
```

Example 3.46 Definition of the phase minimum and the interface thickness

For flow coupling a set of numerical parameters are queried in this section: the solver scheme, convergence criteria, and iteration limits. Section 4.15.4 describes the necessary parameters and ways to arrive at suitable values for a specific simulation.

In case of stress coupling, the convergence criteria for the iterative BiCG-stab matrix solver as well as an upper limit (maximum number) of iterations have to be defined. In the input line for the maximum number of iterations, the input of a second integer is also possible. It is set to 20 by default and specifies the maximum number of iterations allowed for achieving a good solution for quasi-static equilibrium. In most cases, the default settings are sufficient.

Next, a value for the phase minimum is required. A cell with a phase or grain fraction below this value is not considered to be in the interface any more but in a bulk region (liquid or solid). In most cases, a value of 1.E-4 can be recommended.

As optional parameter in the same line, a hysteresis factor can be defined which has a default value of 2, For more information, please consult the MICRESS forum (www.micress.de/forum).

Finally, the interface thickness (in cells) also has to be input in this section. Generally, with increasing interface thickness, the curvature evaluation of the phase-field profile is improved. On the other hand, a higher resolution is required to resolve highly curved structures, and numerical artefacts related to the finite interface thickness like “artificial solute trapping“ is increased.

Note that in case of using extremely small values for the interface thickness (<~3 cells), curvature evaluation is poor and nucleation at the interface is not working correctly anymore!

3.2.21 Parallel execution parameter

The parallel variants of MICRESS® require one additional parameter for the number of threads to be used for parallel execution of the diffusion, stress, and temperature solvers.

These solvers are parallelized using the OpenMP (see www.openmp.org) paradigm for parallel programming which is intended for shared memory, thread level based parallelization.

Driving File

```
...
# Number of parallel threads?
# ====================
3
...
```

Example 3.47 Thread count for accelerating the execution of diffusion, stress, and temperature solvers in parallel MICRESS variants

It is possible to reduce the wallclock time of a simulation if the computational load is put heavily on the parallelized solvers. Check the 'TabP' output for getting an impression in which cases you can benefit from using the parallel variants of MICRESS®, see Figure 3.13. Each line of the text output shows the accumulated time values for the current simulation time step. Except the 'Simulation Time' and 'CPU Time', all values show wallclock time. The third column, headed with 'Wallclock Time', shows the overall execution time and reflects the real time time gone for the simulation.

Delta_Gamma_2.TabP

# Simulation time [s]	CPU Time [s]	Wallclock Time [s]	TQ Time [s]	PF Time [s]	Diff Time [s]	ATC Time [s]	List Time [s]
...							
49.00000	27708.83	22519.44	18175.83	1038.460	2534.988	0.000000	484.8080
49.20010	27975.82	22749.42	18376.18	1046.644	2551.406	0.000000	488.8440
49.40021	28186.28	22921.57	18518.39	1054.909	2567.984	0.000000	492.9190
49.60024	28391.97	23090.58	18657.64	1063.109	2584.481	0.000000	496.9260
49.80008	28595.35	23257.09	18794.47	1071.243	2600.993	0.000000	500.9070
50.00000	28735.55	23361.27	18868.77	1079.570	2617.479	0.000000	504.9760

Delta_Gamma_8.TabP

# Simulation time [s]	CPU Time [s]	Wallclock Time [s]	TQ Time [s]	PF Time [s]	Diff Time [s]	ATC Time [s]	List Time [s]
...							
49.00000	55932.01	21876.79	18112.85	1038.396	2022.776	0.000000	487.5380
49.20008	56396.21	22105.78	18314.82	1046.781	2036.221	0.000000	491.7670
49.40018	56804.83	22273.12	18454.96	1055.233	2049.619	0.000000	496.1910
49.60033	57206.30	22440.08	18594.56	1063.723	2063.111	0.000000	500.5600
49.80002	57612.18	22607.04	18734.23	1072.212	2076.512	0.000000	504.9300
50.00000	57957.71	22710.49	18810.38	1080.738	2089.927	0.000000	509.3020

Figure 3.13 Performance Output TabP for 2 and 8 threads

Recommended choices of MICRESS® variants:

- Serial versions: If there is only little load on the parallelized solvers, it is recommended to use MICRESS® standard serial variants. The parallelization itself introduces some overhead which results in a slow down of the parallel simulation.
- Parallel versions: Simulation with a fair load on the parallel solvers will benefit from parallel execution with thread counts above 1. The performance will depend on the problem size and the available CPUs on your system when you choose high thread counts. Usually, there is an optimal thread count and the performance will degrade beyond.

3.3 How to create an input file?

There are two common ways to prepare an input (driving) file for a given simulation. They are:

- use the text editor to modify an already existing driving file
- create a new driving file interactively, using the "__switch__" ("underline underline switch underline underline") option

Modifying an already existing driving file may be very convenient if the user wants to perform a simulation similar to an already existing one. In this case, specific parameter values can be changed fast while conserving the overall structure of the input file. For more fundamental changes, the _switch_ option may be helpful, which enables the interactive input of an entire block when modifying the input file. It can be applied successfully when the user is not sure about the exact sequence of the input expected by MICRESS®.

For that purpose, the __switch__ keyword is inserted within an existing driving file to mark the new or changed block. When the _switch_ keyword is reached during reading from driving files, the program switches to reading further input directly from keyboard. The interactive input can be ended by a new _switch_ keyword marking the end of the block. Among the result files the user will find the modified input file having the extension ".in". This input file can be used as a new driving file and modified using a text editor.

3.4 Start the calculation

Under UNIX, type the name of the MICRESS® binary followed by the driving file, MICRESS® will then parse its contents. Under Windows, drag and drop the driving file on the MICRESS® binary (or a link to it) or use the context menu by right-clicking the driving file (".txt"-extension). The context menu should have been generated by the installer during the initial installation of the MICRESS® software.

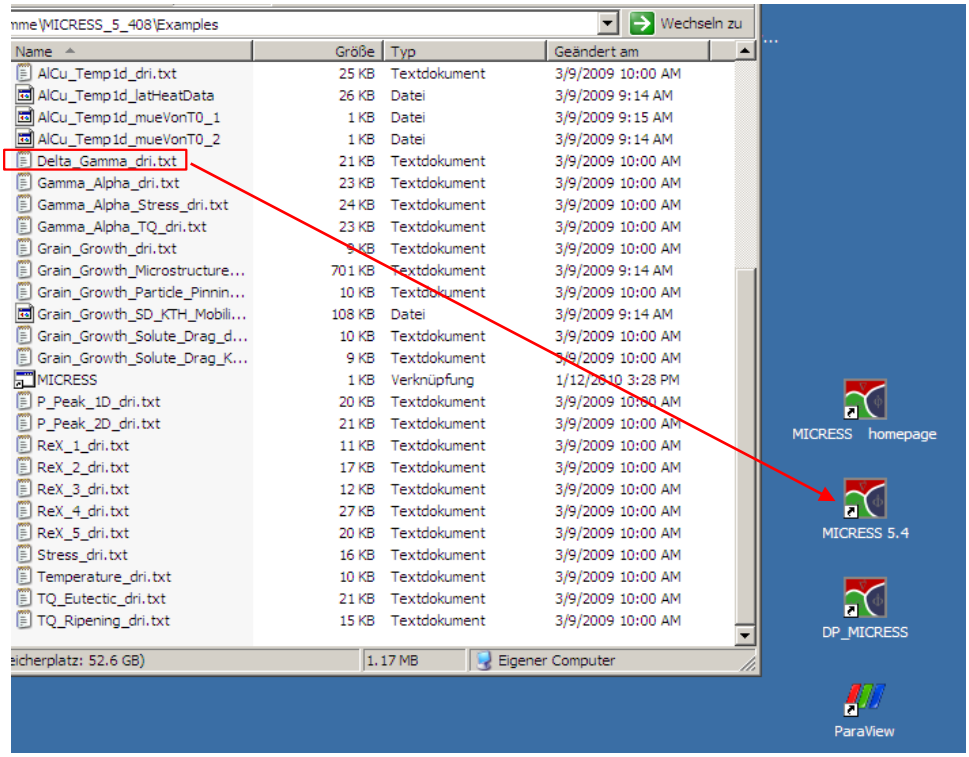

nme\MICRESS_5_408\Examples
Wechseln zu
Name
Größe
Typ
Geändert am
AlCu_Temp1d_dri.txt 25 KB Textdokument 3/9/2009 10:00 AM
AlCu_Temp1d_latHeatData 26 KB Datei 3/9/2009 9:14 AM
AlCu_Temp1d_mueVonT0_1 1 KB Datei 3/9/2009 9:15 AM
AlCu_Temp1d_mueVonT0_2 1 KB Datei 3/9/2009 9:14 AM
Delta_Gamma_dri.txt 21 KB Textdokument 3/9/2009 10:00 AM
Gamma_Alpha_dri.txt 23 KB Textdokument 3/9/2009 10:00 AM
Gamma_Alpha_Stress_dri.txt 24 KB Textdokument 3/9/2009 10:00 AM
Gamma_Alpha_TQ_dri.txt 23 KB Textdokument 3/9/2009 10:00 AM
Grain_Growth_dri.txt 9 KB Textdokument 3/9/2009 10:00 AM
Grain_Growth_Microstructure... 701 KB Textdokument 3/9/2009 9:14 AM
Grain_Growth_Particle_Pinnin... 10 KB Textdokument 3/9/2009 10:00 AM
Grain_Growth_SD_KTH_Mobili... 108 KB Datei 3/9/2009 9:14 AM
Grain_Growth_Solute_Drag_d... 10 KB Textdokument 3/9/2009 10:00 AM
Grain_Growth_Solute_Drag_K... 9 KB Textdokument 3/9/2009 10:00 AM
MICRESS 1 KB Verknüpfung 1/12/2010 3:28 PM
P_Peak_1D_dri.txt 20 KB Textdokument 3/9/2009 10:00 AM
P_Peak_2D_dri.txt 21 KB Textdokument 3/9/2009 10:00 AM
ReX_1_dri.txt 11 KB Textdokument 3/9/2009 10:00 AM
ReX_2_dri.txt 17 KB Textdokument 3/9/2009 10:00 AM
ReX_3_dri.txt 12 KB Textdokument 3/9/2009 10:00 AM
ReX_4_dri.txt 27 KB Textdokument 3/9/2009 10:00 AM
ReX_5_dri.txt 20 KB Textdokument 3/9/2009 10:00 AM
Stress_dri.txt 16 KB Textdokument 3/9/2009 10:00 AM
Temperature_dri.txt 10 KB Textdokument 3/9/2009 10:00 AM
TQ_Eutectic_dri.txt 21 KB Textdokument 3/9/2009 10:00 AM
TQ_Ripening_dri.txt 15 KB Textdokument 3/9/2009 10:00 AM
icherplatz: 52.6 GB)
1.17 MB
Eigener Computer
MICRESS homepage
MICRESS 5.4
DP_MICRESS
ParaView

Figure 3.14 Start the calculation by drag and drop

Windows also allows running MICRESS® in a MS-DOS shell, similarly to UNIX. This way of starting MICRESS® has the advantage that in case of a program break, the execution window is not closed automatically and error messages can be seen more easily.

3.5 Check the output

During run-time, some information can be read on screen. As soon as the input file is dropped on the MICRESS® binary, its contents is displayed, being read by the program at the same time. During the calculation procedure, some further information can be reviewed such as the current time-step, the processing time and the temperature at the bottom of the calculation domain.

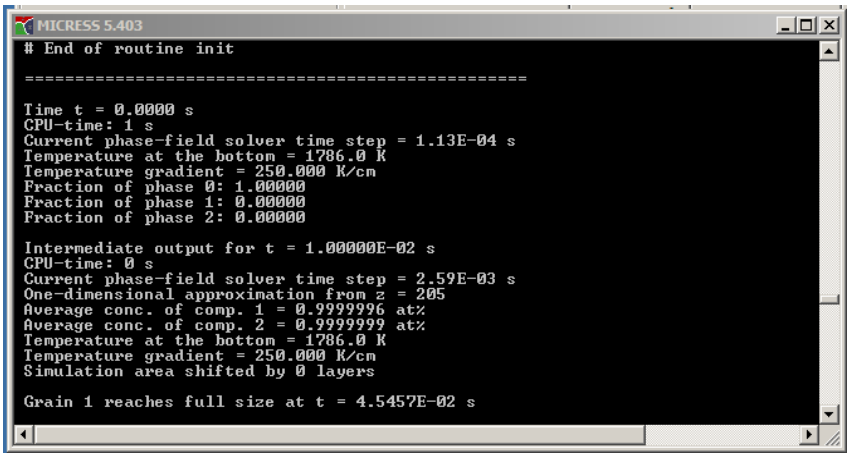

MICRESS 5.403
End of routine init
Time t = 0.0000 s
CPU-time: 1 s
Current phase-field solver time step = 1.13E-04 s
Temperature at the bottom = 1786.0 K
Temperature gradient = 250.000 K/cm
Fraction of phase 0: 1.00000
Fraction of phase 1: 0.00000
Fraction of phase 2: 0.00000
Intermediate output for t = 1.00000E-02 s
CPU-time: 0 s
Current phase-field solver time step = 2.59E-03 s
One-dimensional approximation from z = 205
Average conc. of comp. 1 = 0.9999996 at%
Average conc. of comp. 2 = 0.9999999 at%
Temperature at the bottom = 1786.0 K
Temperature gradient = 250.000 K/cm
Simulation area shifted by 0 layers
Grain 1 reaches full size at t = 4.5457E-02 s

Figure 3.15 The MICRESS command line: initialisation and start of the simulation

In addition, MICRESS® issues a message summarising information every time a new phase or grains appears such as seed number and nucleation time.

As already mentioned, the MICRESS® output consists of various files, whose content is specified by the extension, e.g.:

- *.log is the log file of simulation-time output (in ASCII).
- *.in is the driving file created by MICRESS® (which can be used as a new driving file).
- *.geoF stores information relative to the geometry (i.e.: grid size, cell dimension).

MICRESS on-screen output

Seed number 5 set at time t = 1.00E-01 s
---
at a triple-junction
Phase: 2
Seed type: 1
Temperature at the bottom = 1099.0 K
Undercooling = 50.900 K
Grain number = 14

Figure 3.16 MICRESS on screen notifying message when a new phase/grain appears

- *_TQ.txt (Windows specific) stores: the output of Thermo-Calc™´s TQ interface which makes it more convenient to see the text of TQ 'fatal' errors

The post-processing tool DP_MICRESS enables the visualisation of the graphical MICRESS® outputs (see the picture below). Detailed information about post-processing the MICRESS® simulation can be found in the MICRESS® User's Manual Volume III: MICRESS® post-processing.

Important: MICRESS® Windows version creates file names in a slightly different way: the previous extensions are appended to the simulation name with an underscore, and then followed by ".mcr" for binary files and ".txt" for ASCII files. The reason for this is to be able to make use of the Windows file association features. For example,

the UNIX file “Simulation.log“ becomes “Simulation_log.txt“ and “Simulation.phas“ becomes “Simulation_phas.mcr“. The conversion tool from Display MICRESS can also be used to rename the files once the simulation has been completed.

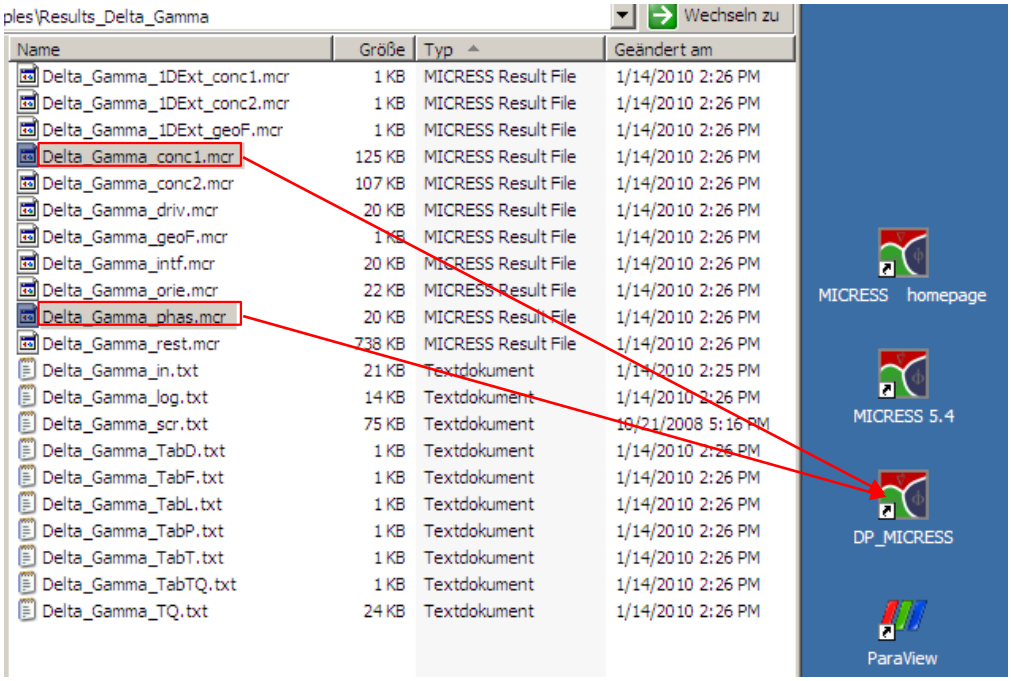

ples\Results_Delta_Gamma
Wechseln zu
Name
Größe
Typ
Geändert am
Delta_Gamma_1DExt_conc1.mcr 1 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_1DExt_conc2.mcr 1 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_1DExt_geoF.mcr 1 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_conc1.mcr 125 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_conc2.mcr 107 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_driv.mcr 20 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_geoF.mcr 1 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_intf.mcr 20 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_orie.mcr 22 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_phas.mcr 20 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_rest.mcr 738 KB MICRESS Result File 1/14/2010 2:26 PM
Delta_Gamma_in.txt 21 KB Textdokument 1/14/2010 2:25 PM
Delta_Gamma_log.txt 14 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_scr.txt 75 KB Textdokument 10/21/2008 5:16 PM
Delta_Gamma_TabD.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TabF.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TabL.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TabP.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TabT.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TabTQ.txt 1 KB Textdokument 1/14/2010 2:26 PM
Delta_Gamma_TQ.txt 24 KB Textdokument 1/14/2010 2:26 PM
MICRESS homepage
MICRESS 5.4
DP_MICRESS
ParaView

Figure 3.17 Visualisation of the results by drag and drop at DP_MICRESS under Windows

4 Further Useful Information

4.1 Considerations about 2D simulation

Real microstructures typically are 3D. MICRESS® can treat all. 1D, 2D and 3D problems, but, naturally, 3D simulations are most time-consuming. For this reason it is worth checking whether a specific 3D problem can be reduced to an equivalent 2D or 1D problem with sufficient accuracy.

Some problems are inherently 3D, like e.g. fibrous eutectic growth or fluid flow through a dendritic mushy zone. But in many other cases, a 2D simulation with high spatial resolution can give more exact information than a poorly resolved 3D simulation. Therefore it is important to think about the consequences of a 2D approximation and to check how they can be eventually corrected.

In general, a 2D calculation domain can be constructed as:

- a section through the structure. As a consequence, however, if there are various small particles like in Figure 4.1, only a few of them will be found in the 2D cut. The mean distance between the particles will not be correct and the interaction between the diffusion fields will not be described well. Thus, this is not a good option for microstructure simulation, although experimental micrographs typically are like that!

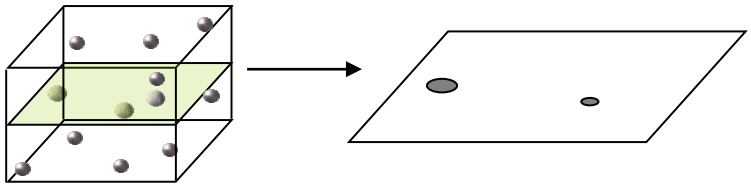


Figure 4.1 Graphical representation of a 2D section

- a projection of the 3D structure onto 2D (see Figure 4.2) This will result in correct average particle distance, but in wrong average phase fractions. The release of latent heat, if used in the simulation, will be predicted incorrectly. In early solidification stages when the solid fraction is small and growing grains are far apart, solutal interactions between the structures can be neglected. Then an analytical correction of the phase fraction of the growing phase is possible. In case of latent heat, this correction needs to be done already during simulation; it is done automatically if the option “latheat_3D“ is selected. For more information see [1].

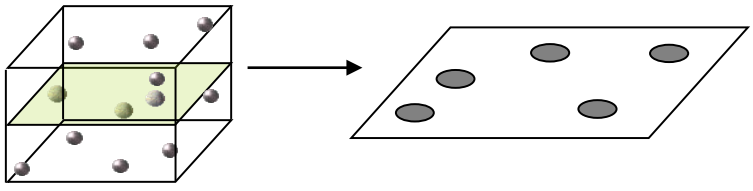


Figure 4.2 Graphical representation of a projection of a 3D structure onto 2D

4.2 Driving force

4.2.1 Driving force options

The driving force options in the phase-interaction input allow increasing numerical stability and (together with the interface mobility and the antitrapping model) calibrating the correct interface kinetics. They consist of parameters

on averaging, limiting and (in future) noise.The optional driving force parameters “avg“, “max“ and “smooth“ are defined as follows:

“avg“:

The averaging factor stands for the intensity of averaging or a relative length over which averaging is done. It is also meant to help stabilizing the interface. This allows the user to move continuously from no averaging to complete averaging of the chemical driving force along the interface normal. Averaging the driving force avoids spreading of the interface, which can occur if the resolution is too low and there is a strong gradient of the driving force over the interface. On the other side, averaging increases artificial solute trapping. The factor is defined as the averaging length over the sum of the interface thickness and the averaging length. The allowed values range from 0 to 1 (no averaging to complete averaging), a typical value that works in most cases is 0.5.

"max":

This parameter allows specifying an upper limit for the driving force in J/cm3 which increases numerical stability. One should choose a value which is above what is to be expected to be on the save side (e.g. 1000). A smoothing of the cut-off is done using a tanh-function. Do not define it if this is not intended! Its application can be useful because of the following reasons:

- While using TQ coupling, there is always a risk of having local numerical instabilities which can occasionally result in "astronomically" high values of the driving force. These, in turn, produce astronomical increments of the phase-field variable, etc. This is a good reason to set a (less than the astronomical) limit to the driving force!
- The DP_MICRESS output of the driving force gets nicer (without the need to rescale), if the maximum driving force is in the same order of magnitude as the majority of the values.
- Sometimes, (e.g. during the initial transient or after nucleation), the intermediately forming microstructure cannot be resolved properly at a given scale. Limiting the driving force gives a chance to reduce numerical trouble without having to change the phase-field mobility or to use a higher grid resolution.

"smooth":

The parameter "smooth" is a "directional noise" on averaging which can be helpful for reducing the effect of grid anisotropy, i.e. a directional averaging is obtained. The local phase-field gradient is rotated randomly with a maximum amplitude of the given angle (in degrees), in 3D along three axes. This procedure is done for each interface cell independently, i.e. for each cell a complete path through the interface is calculated

4.2.2 Averaging of the driving force

Averaging of the driving force is important for reducing the effect of the artificial solute trapping which is a problem for all phase-field methods when the spatial resolution is too low.

It is possible to fine-tune averaging by specification of an averaging factor avg. The effect of a high value of this factor is a strong reduction of the risk of an interface spreading (interface gets thicker than the given value of η), but at the expense of a reduced effective mobility (i.e. the front moves slower because the average solute undercooling is decreased) and at the expense of an increased grid anisotropy (e.g. dendrites may tend to follow the grid direction instead of the anisotropy direction). In most cases, it seems to be best to use values around 0.5 which corresponds to an "averaging length" of the order of the interface thickness. For fine-tuning of the kinetics, both the interface mobility and the averaging length have to be optimized (see "Numerical parameters").

How does the averaging method used by MICRESS® work for the calculation of the average driving force? The chemical driving force ΔG for each interface grid point is averaged along the direction of the normal through the interface. The weight of each cell in the averaging process consists of 3 factors:

1. the length of the path along the normal through the given cell (i.e. cells which are just touched slightly by the normal direction have less weight). This is important to avoid possibly strong fluctuations of the driv-

ing force if the normal direction changes slightly. “Locking“ the averaging path to the main grid directions can be avoided by an extra noise on the normal direction using the “smooth“ keyword.

2. the gradient term (i.e. effectively by the term SQRT(ϕ_1*ϕ_2) for the double obstacle potential) in the cell along the gradient path.
3. according to the avg input which is transformed to an averaging length l: l = avg/(1 - avg)*η where η is the interface thickness. The distance for averaging is restricted to the distance l from the interface grid point for which the averaged driving force is calculated. Inside the averaging region the corresponding weighting factor is proportionally decreasing from 1 in the centre to 0 at the edge.

For the extreme cases of avg = 0 and avg = 1 one gets no averaging at all or complete averaging over the entire interface length including the weighting according to 1 and 2. The cut-off of the driving force (“max“ keyword) is done **before** the averaging.

4.2.3 Driving force calculation using linearised phase diagrams

Using the keyword "linear" in the phase diagram input, one can define a linear phase diagram like the one shown below, where the two phase lines may intersect for c = 0. In the general case, the phase can intersect at any concentration c ≠ 0.

Important: the temperature T_0, for which the linearisation parameters are specified in the input file, and the corresponding equilibrium compositions are not shown here.

At the applied temperature T, the interface most probably is not in equilibrium, because the local phase fractions (phase-field parameter) inside the interface (i.e. for each interface cell) are not in accordance with the mixture composition c and the equilibrium compositions c_L*/c_S* for this given temperature T.

Instead, accordance can be found for another temperature T_{eq}, which can be determined easily for a binary phase diagram, but which can also be calculated straightforward for multicomponent systems.

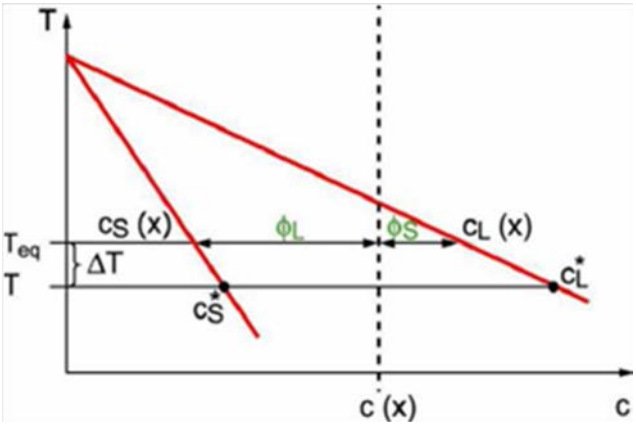


Figure 4.3. Calculation of the driving force using linearised phase diagrams

Thus, the effective equilibrium compositions are taken at the temperature T_{eq} rather than at the real temperature T. The difference between T and T_{eq} is used to calculate the chemical driving force: $\Delta G = \Delta S * \Delta T = \Delta S (T - T_{eq})$

In the multiphase case, T_{eq} is determined independently for each pair-wise phase interaction. Therefore, in triple junctions each pair-wise interface can be close to or far from local equilibrium. In the case of Thermo-Calc™ coupling as well as for "linearTQ", the construction of the (extrapolated) linear phase diagram is slightly different. The diagram is constructed rather in terms of ΔG than of T, the slopes m' of the phase lines are derived from the driving force ΔG and related to the "real" slopes m:

$m'_{1/2} = \delta(\Delta G)/\delta c_1 = \Delta S_{1/2} * m_{1/2}$
$m'_{2/1} = \delta(\Delta G)/\delta c_2 = \Delta S_{2/1} * m_{2/1}$

Furthermore, a driving force offset and a temperature dependence of the equilibrium concentrations $\delta c/\delta T$ is calculated or can be specified.

4.3 Outputs

Most MICRESS® results are binary sequential files written by the Fortran MICRESS® code. They can be opened and processed directly using DP_MICRESS. Some other files have ASCII format and thus can be opened by any text editor.

If other post-processing software is to be used, some considerations about the file format are necessary:

If read by a user program written in another language, the “record length“ should be adjusted: it is a 4-byte integer (the length of the record in bytes).

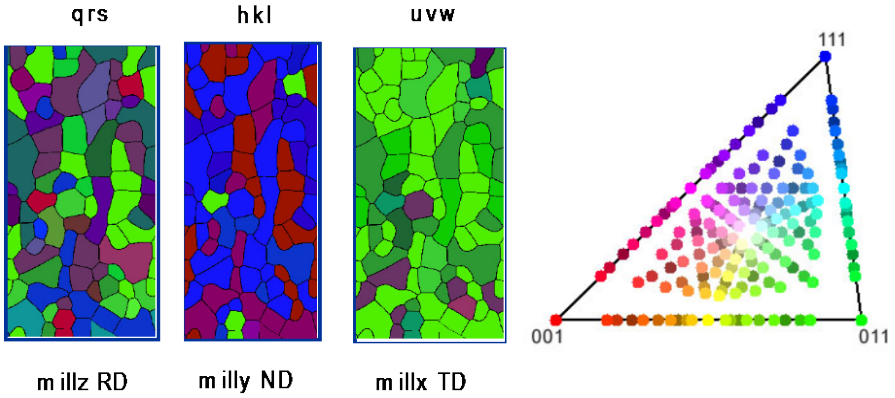
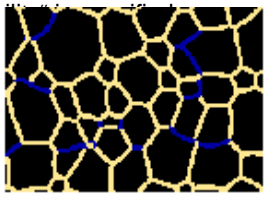
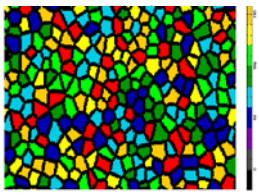
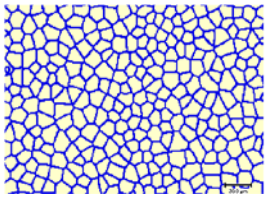
The output files are written in a “native“ format, i.e. “little-endian“ on platforms using Intel-compatible processors. They may be “big-endian“ in other cases.

The result files contain the field value for a given output-time, one record for each. First, the time (4-bytes real) is written, then the number of points in the simulation domain (a 4-bytes integer which is the product of the number of points in the x, y and z-directions), followed by the field values for each cell (a “number of points“, 4-byte integers). They are ordered in the following way: first all the values for x, y=1 and z=1, then the same for y=2 (if applicable), for z=2 and so on.

4.3.1 List of available outputs

The various available MICRESS® output files are listed in alphabetic order according to their extensions with a brief description of each. They are written only if the individual output type is requested in the driving file and if the output type applies to the chosen type of coupling (e.g. .conc{k} output is written only in case of concentration coupling)

*.conc{k}	Gives information about the concentration of the component k if concentration coupling is used. The concentrations are in at % or wt % (as set in the driving file).
*.c{k}Pha{i}	Gives the composition of the component k in the i-phase. In the interface region between the phase i and other phases, only the phase composition is written (as opposed to the mixture concentration).
*.dGsp	Used in stress-coupled simulations. It shows the stress contribution to the driving force (if “out_dG_stress“ is chosen), in J/cm3.
*.dTLat:	Contains data related to the release of latent heat (if concentration coupling and latent heat (if “lat_heat“ or “lat_heat_3d“ keyword is activated), in ASCII format.
*.diff	Shows diffusion data in ASCII format if Thermo-Calc™-coupling is activated and mobility databases are used.
*.driv	Gives the driving force for the growth of a grain with a higher number into a lower one (if “out_driv_force“ is set), in J/cm3. It displays only the chemical part of the driving force and the driving force for recrystallisation. pair-wise
*.frac{i}	Contains the fraction of phase i (if “out_fraction“ is specified). Several phases can be written simultaneously. If no phase number is specified, the sum of all solid fractions is written.
*.geoF	Holds information concerning the geometry (i.e.: grid size, cell dimension). This file is needed for displaying binary files using DP_MICRESS.

*.hStr	Provides information about the hydrostatic stress in MPa, if stress coupling is activated and “out_vMises_stress“ is specified.
*.in	Represents a clean ASCII driving file created by MICRESS®. It can be used keeping the driving file in a readable form.
*.intf	Displays the interface region with values of -1 for dual interfaces, -3 for triple junctions, -6 for quadruple junctions etc. (if “out_interface“ is specified).
*.korn	Displays the grain number, if “out_grains“ is specified“. The grain number is not a physical property, but allows for the identification of grains in the MICRESS® results. In contrary to the .phas-output, the .korn file does not show interface regions. In the interface region, the number of the grain with the highest fraction is given.
*.krum	Shows the interface curvature in µm-1, if “out_curvature“ is specified.
*.millx, *.milly, *.millz	Shows the respective Miller indices, when “out_miller“ is specified“. 
*.mueS	Displays the local interface mobility in cm⁴/(Js), if “out_mo 
*.orie	Shows the 2D-grain orientation in degrees, if “out_orientati 
*.phas	Gives information about the phase number, if “out_phase the interfaces, why it can be also used for displaying gr growth. 
*.rest	Contains data necessary to restart a simulation. It is written if “out_restart“ is specified and contains only data from the most recent output time. This binary file is for internal use by MICRESS® and cannot be post-processed.

*.rex	Shows the stored energy in J/cm^3, if “out_recrystall“ is specified.
*.rhoD	Contains the dislocation density field [m^{-2}] if “out_disloc” is specified. Avaliable only if the local RX energy model or the mean variant with specified mean dislocation density is adopted.
*.sxxCV	Also *.sxyCV, *.sxzCV, *.syyCV, *.syzCV and *.szzCV components of the stress tensor, xxCV, (yyCV), zzCV are the diagonal elements in the 2D (3D) case, sxyCV, sxzCV, syzCV are off-diagonal terms (torsion), respectively. The output is available if the stress coupling is activated and “out_normal_stress“ is specified; in MPa.
*.TabC	Contains the average and extrema of the concentration in each phase in ASCII format, if the concentration coupling is activated and “tab_conc“ is specified.
*.TabD	Gives the diffusion coefficients (only diagonal terms) in all phases at the corresponding “tab_log“ output times in ASCII format, if concentration coupling is activated and „tab_log“ is specified.
*.TabF	Gives the average fraction of each phase in the whole domain, if the concentration coupling is activated and ’“tab_fractions“ is specified; in ASCII format.
*.TabGD	An ASCII file which contains information about the grain, e.g. diameter, neighbours (if ’“tab_grain_data“ is specified).
*.TabK	Contains the number of grains in ASCII format, if “tab_grains“ is specified. This file is updated by default each time a grain is set or disappears.
*.TabL	This is an ASCII-monitoring output, if “tab_log“ is specified. This monitoring output is generated at user-defined intervals to check the simulation progress; it also contains additional information that depends on the type of coupling activated.
*.TabN	Stores data in ASCII format, relevant for making “Von Neumann – Mullins plots“, if “tab_vnm“ is specified.
*.TabR	An ASCII file for the recrystallized fraction, if “tab_recrystall“ is specified.
*.TabO	An ASCII file for orientation information, if “tab_orientation“ is specified.“
*.TabP	An ASCII file which contains performance information for several groups of routines.
*.TabT	Reports information in an ASCII file concerning the setting of the (automatic) time-step, if automatic time-stepping has been selected and “tab_log“ specified. ASCII format.
*.temp	The temperature in Kelvin, if “out_temperature“ is specified. It can be also used for plotting temperature-distance profiles. For alloy simulation (i.e. in the case of concentration coupling), the temperature changes only in z-direction, so temperature-distance profiles should be plotted along the z-axis.
*_TQ.txt	A Windows-specific data file. It stores the output of Thermo-Calc™´s TQ interface which makes it more convenient to see the text of TQ fatal errors. This file is always created under Windows when Thermo-Calc™ -coupling is activated.

*.uxCV, *.uyCV, *.uzCV	Records the normal displacements if the stress coupling is activated and "out_normal_displacement" is specified. in percent.
*.vel	For the interface velocity if "out_velocity" is specified. in cm/s.
*.vxCV	Also *.vyCV, *.vzCV, *.syyCV, *.syzCV and *.szzCV: components of the stress tensor; xxCV, (yyCV), zzCV are the diagonal elements in the 2D (3D) case, sxyCV, sxzCV, syzCV are off-diagonal terms (torsion), respectively. This is true if the stress coupling is activated and "out_normal_stress" is specified; in MPa.
*.vM	For the von Mises stress if stress coupling is activated and "out_vMises_stress" is specified; in MPa.

4.3.2 How to use the driving force output

The driving force output can be considered as one of the most important outputs. In the following the meaning of the driving force output will be explained using a concentration coupling example.

The driving force output is a field output which gives the value of the chemical driving force ΔG at the interface region. To be exact, it is just the "bulk" of the interface, i.e. not the cells which have a phase fraction of zero in one phase but still are part of the interface list. For that reason, the interface region appears somewhat broader in the .intf output than in the .driv output. In triple or higher junctions it is impossible to show all the driving forces for the pair-wise interactions and only the 'highest' one is displayed (e.g. for phase pair 1/2 if you have a triple junction built up from grain 0, 1 and 2, with the pair-wise interactions 0/1, 0/2 and 1/2).

ΔG can be related to the undercooling ΔT by dividing by the transformation entropy ΔS, it is written e.g. in the .log -file during initialisation in case of Thermo-Calc coupling.

Due to the fact that the chemical driving force is a function of the local concentrations, and the local concentrations typically are close to local equilibrium conditions, the curvature effects may be visible in the .driv-output. In fact, the observed local (solutal) undercooling $\Delta T=\Delta G/\Delta S$ is always the sum of a kinetic and a curvature contribution (if no other contributions like stress etc. are present):

$$\Delta T = \Delta T_{Kin} + \Delta T_{Curve}$$

The information obtained from the .driv-output may be summed up as follows:

- if the interface mobility is high, ΔT_{Kin} is very small and the curvature undercooling (or more exactly the driving-force which corresponds to curvature) may be seen directly. If no averaging is used, ΔG will be only zero in average but there can be still non-zero values with opposite sign in the inner and outer side of the interface.
- If the mobility is low, ΔT_{Kin} may be much higher than ΔT_{Curve}, then one will see mainly the kinetic contribution to ΔG.
- If the interface is not curved in some regions of the simulation, the .driv file will show only the kinetic contribution in these regions. As a general rule for diffusion-coupled growth (if no exact calibration of the interface kinetics is intended), the kinetic undercooling should be in the order of or less than the curvature undercooling, but this of course depends always on the individual simulation setup. A good indication of a too small mobility value is if driving force values increase steadily with simulation time although a constant cooling rate is applied. If conditions vary strongly within one simulation (e.g. if the simulation spans a wide temperature range) and a temperature-dependent mobility is used, it will be hard to reasonably adjust the mobility function without monitoring the .driv-output along the simulated process path.

- If a too high mobility is used, the interface may get unstable. This instability can be seen first as a freckling and/or fluctuation of the .driv-output in the corresponding interface which indicates fluctuating phase-field increments. In general, any kind of instability typically can be first detected in the .driv-output.

As a conclusion, the .driv-output is essential for adjusting interface kinetics and for getting numerically stable simulations.

4.4 Time input data

4.4.1 Automatic time stepping

The use of automatic time stepping is strongly recommended in most applications. The number of numerical parameters and the risk of numerical instabilities are reduced. Optimum performance is achieved by automatically choosing the highest stable time-step value.

There are two independent criteria for automatic time stepping: a pure "phase-field criterion" and a segregation criterion. Both are stability criteria. The simulation usually gets numerically unstable if a too large time step is used.

According to the phase-field criterion, the time step must not exceed a critical value, depending on the spatial resolution Δx, the surface energy σ and the interface mobility μ. The segregation criterion is only relevant in case of concentration coupling and depends on the redistribution thermodynamics.

For both criteria, a pre-factor can be defined as optional parameter in order to allow manipulation, a value of 1.0 exactly corresponds to the criteria shown above.

The resulting effective phase-field time-step as well as the time-step values of the individual criteria can be monitored in the .TabT-output. Due to the fact that a "worst case scenario" is used for the segregation criterion, in some cases a pre-factor >1 may be still acceptable and could even increase substantially the performance of the whole simulation.

The drawback of the "worst-case scenario“ for the segregation criterion is that the behaviour of one grid cell may affect the time-step and thus the performance of the whole simulation. If one single interface cell experiences a numerical problem and combined with unreasonable thermodynamic parameters, this single cell could extremely decrease the global time-step. This can practically freeze a simulation. The solution is the definition of a minimal time-step which can be specified as a third optional parameter to the “automatic“ keyword. If a minimal time step is used, the interface mobility μ of those (exceptional) cells which would need a smaller time-step than the minimal one is reduced. One should check the .mueS- output to make sure that this does not happen for too many cells.

4.5 Phase data

4.5.1 Anisotropic phase definition and crystallographic symmetry

The growth of a solid phase may strongly be affected by its underlying atomic lattice structure. Especially the interfacial mobility and the interfacial energy are commonly functions of orientation. In order to consider crystallographic aspects of a solid phase it first has to be defined as non-“isotropic”, i.e. as either “anisotropic” or “faceted” (faceted_a, faceted_b, antifaceted). This is the basic requirement to define crystallographic orientations and anistropic phase-interactions.

Moreover, a crystallographic symmetry can be defined for this phase by selecting one of the following options: ”cubic”, “hexagonal”, “tetragonal” or “orthorhombic” as illustrated in Table 4.1. The most common type of anisotropy in metals is the cubic anisotropy applicable to fcc and bcc phases and also to diamond structures. The hexagonal symmetry may be used e.g. for simulation of Mg or Zn alloys. Note that in MICRESS®, the hexagonal symmetry is defined with the basal direction c parallel to y such that the six-fold basal plane corresponds to the x-z-

plane which is used as standard in 2-d simulations). Tetragonal symmetry and orthorhombic symmetry are both distorted cubic structures defined with the c-axis parallel to the z-axis of the grain coordinate system. If none of these symmetry types are applicable simply select “none”. The definition of the crystallographic symmetry is required for the following tasks:

- to reduce grain orientations to equivalent orientations in the fundamental zone,
- to calculate the disorientation between neighboring grains, i.e. the equivalent misorientation with smallest angle,
- to determine the anisotropy functions for the interfacial energy and the interfacial mobility in case of weak anisotropic phase interactions.

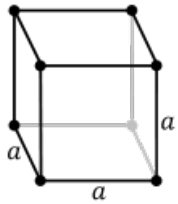
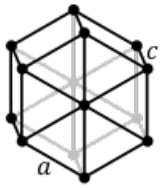
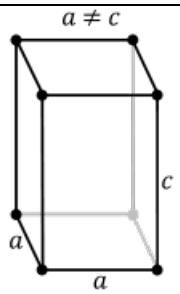
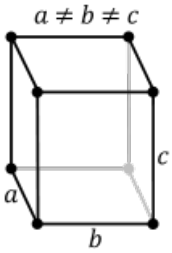
			
cubic	hexagonal	tetragonal	orthorhombic

Table 4.1 Crystallographic symmetries implemented in in MICRESS

4.5.2 Faceted growth

In addition to the standard weak anisotropy model, MICRESS® offers special anisotropy models for faceted growth. These models can be selected by defining a solid phase as either "faceted_a", "faceted_b" or "antifaceted".and by specifying individual normal vectors of the facets in the local coordinate system. Additionally, a parameter *kappa*, which defines the broadness of the anisotropy function has to be specified. Facets with equal properties can be grouped into a facet type. To treat phases with more than one facet type an amended model version "faceted_b" is recommended which better selects the facet type with lower energy. An alternative model to simulate plate-like growth is given by the "antifaceted" model which in contrast to the faceted model describes fastest growth in direction of the facet vectors.

In the phase interaction input, a static and a kinetic anisotropy coefficient have to be specified for each facet type, if one of the two interacting phases is faceted. The anisotropy of the interfacial energy is implicitely taken into account by the anisotropy of the interface stiffness which allows a direct matching to the modified Gibbs-Thomson equation for anistropic interfaces, which reads: $v=\mu\,[\Delta G-\sigma^*\,\kappa]$ with v being the velocity in normal direction, μ the kinetic coefficient, ΔG the driving force, κ the curvature and σ^* the interface stiffness.

Driving File

```
...
# Data for phase 1:
# ------------------------
...
# Is phase 1 anisotrop?
# Options: isotropic  anisotropic  faceted_a  faceted_b  antifaceted
faceted_b
# Crystal symmetry of the phase?
# Options:   none  cubic  hexagonal  tetragonal orthorhombic
none
# Number of type of facets in phase 1
1
# kin. anisotropy parameter Kappa?
# only one value for all facets/phases
# 0 < kappa <= 1
0.5
# Number of possible orientations of a facet 1
2
#  1 -th normal vector facet  1 ?  3*
1.0
0.0
1.0
#  2 -th normal vector facet  1 ?  3*
1.0
0.0
-1.0
...
# Phase interaction data
# =================
# Is interaction isotropic?
# Optionen: isotropic  anisotropic
anisotropic
# static anisotropy coefficient of facet 1 (0 < a <= 1,
#    1=isotrop, 0 not define d)
0.1
# kinetic anisotropy coefficient of facet 1 (0 <= a <= 1, 1=isotrop)
0.1
```

Example 4.1 MICRESS input for faceted growth

4.5.3 Definition of crystallographic orientations

Crystallographic orientations have to be defined for each grain of an anisotropic. A grain orientation is generally defined as relative rotation between the local grain coordinate-system and the global reference coordinate-system. In 2-d simulations, orientations can simply be described by a single angle, but the description of orientations in 3-d is complex and no common standard exists. Therefore, MICRESS® offers various ways to formulate 3-d orientations:

- Euler angles:

  According to the Euler theorem, any two independent orthonormal coordinate frames can be related by a sequence of three rotations about coordinate axes, where no two successive rotations may be about the same axis. Be careful when importing Euler angles, because there are 12 different representations! MICRESS® uses the ZXZ' convention i.e. the Euler angles phi, theta, psi.

  - rotation 1 (φ): (anticlockwise) about the [001] axis; Z1.
  - rotation 2 (θ): (anticlockwise) about the (rotated) [100] axis; X.
  - rotation 3 (ψ): rotate axes (anticlockwise) about the rotated [001] axis; Z2.

- Angle/Axis:

  According to the definition, any two independent orthonormal coordinate frames can be related by a single rotation with an angle φ about a specific axis (a_z, a_y, a_x).

- Miller indices:

  They are the reciprocals of the fractional intercepts which between the planes and the crystallographic X, Y, Z axes of the three non-parallel edges of the cubic unit cell

Note that if a crystallographic symmetry has been defined, MICRESS® may replace a user defined orientation by an equivalent orientation in the fundamental zone. In case of crystal symmetry, there exist mathematically distinct orientations which are physically indistinguishable. In 3-d, there exist 24 equivalent orientations for cubic symmetry, 12 for hexagonal symmetry, 8 for tetragonal and 4 equivalent orientations for orthorhombic symmetry. For a unique representation of orientations, the full orientation space can be reduced to a fundamental zone. In MICRESS®, the fundament orientation is defined as the crystallographically equivalent orientation with smallest rotation angle $|\varphi|$ and $a_z > a_y > a_x$, with (a_x, a_y, a_z) being the components of the rotation axis.

Symmetry	none	cubic
fundamental zone	0°-360°	0°-90°
orientation α:	30°	30°
orientation β:	80°	80°
orientation γ:	120°	30°
misorientation $\alpha\beta$:	30°/80° → 50°	30°/80° → 40°
misorientation $\alpha\gamma$:	30°/120° → 90°	30°/30° → 0°
misorientation $\beta\gamma$:	80°/120° → 40°	80°/30° → 40°

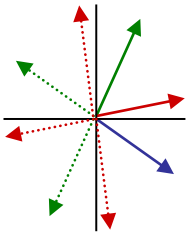


Table 4.2 The orientation spectrum is reduced to the fundamental zone

4.5.4 Categorization

The keyword “categorize“ is used in the MICRESS® driving file with two meanings:

- merging several grains into one
- concentrating continuous orientations into a few distinct orientations

The main reason why categorization has been introduced in MICRESS® is the performance increase which can be achieved by reducing the number of effective grains. As soon as the number of grains exceeds ~1000, the performance of MICRESS® decreases noticeably due to the administrative effort of the interface list structure (Figure 4.4). At the present stage, it gets very hard to get beyond ~10000 grains due to the high memory usage. In many simulation cases, the number of grains or precipitates can get very high, while the individuality of each-single grain is of minor importance, as e.g. in the case of precipitation of a secondary phase. Then, a huge performance gain can be obtained if several grains can be "categorized" in the sense that they are treated as one single grain. One prerequisite is that they have identical properties, i.e. the phase, orientation, recrystallisation energy etc. Furthermore, if they have been nucleated as "small grains", they need to have already reached "full size", otherwise they still have their individual analytical or stabilized curvature and hence cannot be categorized.

The categorization option is controlled in the following way:

- Specifying "categorization" in the phase data allows switching on/off this option for the given phase. All grains of this phase will be checked during runtime with respect to the possibility of assigning them a common grain number.

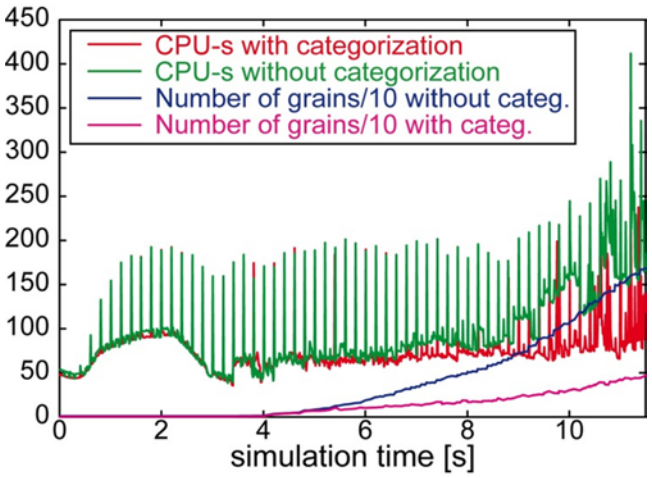


Figure 4.4 Graphical representation of the corelation between categorization and CPU time

- If "categorization" is activated for a given phase which has been defined as anisotropic, one is prompted to switch on/off "categorisation" for each grain or seed type within this phase in both the initial grain setting and the nucleation input. This “categorize“ flag allows the user to arrange the random nuclei orientations in categories, e.g. 10°, 20°, 30°, etc. The number after the “categorize“ keyword is the number of categories, if no number is given, the default value of 36 (for 36 categories with an interval of 10 degrees in 2D) is used. During simulation, grains of the same orientation category can be "categorized" to the same grain number. In the case of isotropic phases, the number of categories is meaningless, because all grains can be treated as if they had the same orientation.

The effects of categorization are the following:

- the grains lose their identity, they appear with the same grain number in the ".korn" output
- the grains will not form an interface between each other if they touch afterwards due to further growth
- categorization will not apply to nuclei that already touch each other, that means that no existing interfaces will be removed
- the grain statistics output (mean radius etc.) will be meaningless or at least difficult to interpret
- if global TC coupling is used (MICRESS 5.404 and above), the approximation of one set of linearisation parameters per grain may be worse if this "grain" exists in several places
- the orientation distribution may get less smooth if one throws orientations into categories (see below)
- The results should not be affected apart from the choice of discrete grain orientations, the special case of global TC-coupling and the grain-related statistic outputs.

4.6 Grain input

4.6.1 Stabilisation and analytical curvature

If a particle is smaller than the numerical interface thickness, the phase-field algorithm is not suitable to evaluate the correct curvature of the particle. This is a problem for nucleation, because new grains upon nucleation are typically much smaller than the typical microstructure length scale. Inoculants which are often added to commercial alloys for grain refinement are usually much smaller than the numerical interface thickness (< 1 µm). If after nucleation the grain diameter (grains are assumed to start growing on a spherical seeding particle with the same radius) is much smaller than the grid spacing, the grains are treated using an analytical expression for the curvature [1]. The "analytical curvature" feature uses a full equivalent curvature instead of the phase-field curvature.

As already explained in Section 3.2.10, using the “stabilisation“ feature, a reduced curvature is assumed at a given stage instead of the phase-field curvature. When the “stabilisation“ option is used, the curvature term is neglected until a sufficient size is reached and curvature can be turned on continuously. Typically, the central cell of the new grain needs at least a fraction larger than 0.5 to be able to be grown by the phase-field algorithm (for triple junctions it may be more complicated). In MICRESS®, the curvature increases linearly with the fraction of the central cell, i.e. the full curvature is experienced not before the grain reaches "full size" (the central cell has a fraction larger than 1-phMin).

The stabilisation option assumes a growing seed. If the critical radius is larger than the grid spacing, the seed will neither grow nor vanish, i.e. it is metastable because of the artificial curvature reduction. This stabilisation may be terminated by using the "kill_metastable" flag. To avoid this situation (due to a very small grid spacing, high interfacial energy or small stored energy), the option "analytical_curvature" should be used.

4.6.2 Voronoi construction

The Voronoi construction is a weighted construction of an initial grain structure. The radii are used as weighting factors for the distances. A standard Voronoi construction is obtained by equal radii which lead to straight boundaries. The curved boundaries which can be achieved by choosing slightly different radii are often more realistic (Figure 4.5), especially to define initial structures for grain growth simulations (see Figure 4.6)

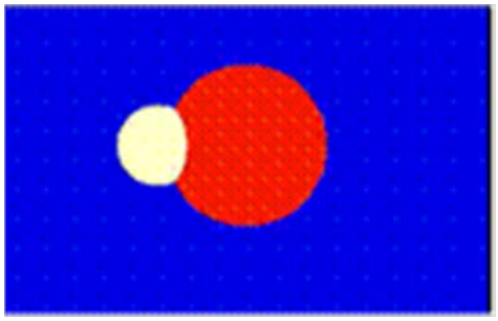


Figure 4.5 The Voronoi construction

During grain ripening, the triple junctions are moving according to their deviation from the equilibrium angle, and the grain boundaries follow their movement driven by curvature. Whether one starts a simulation with a weighted Voronoi construction (curved boundaries) or with a simple Voronoi construction (straight boundaries) depends on the initial assumptions: if one assumes that the initial structure is formed by growth (e.g. recrystallisation), then the simple construction is more appropriate. On the other hand, if it is assumed that some ripening of the structure has already occurred, but this initial ripening shall not be simulated explicitly, one could use the weighted construction model to be closer to a typical steady-state microstructure.

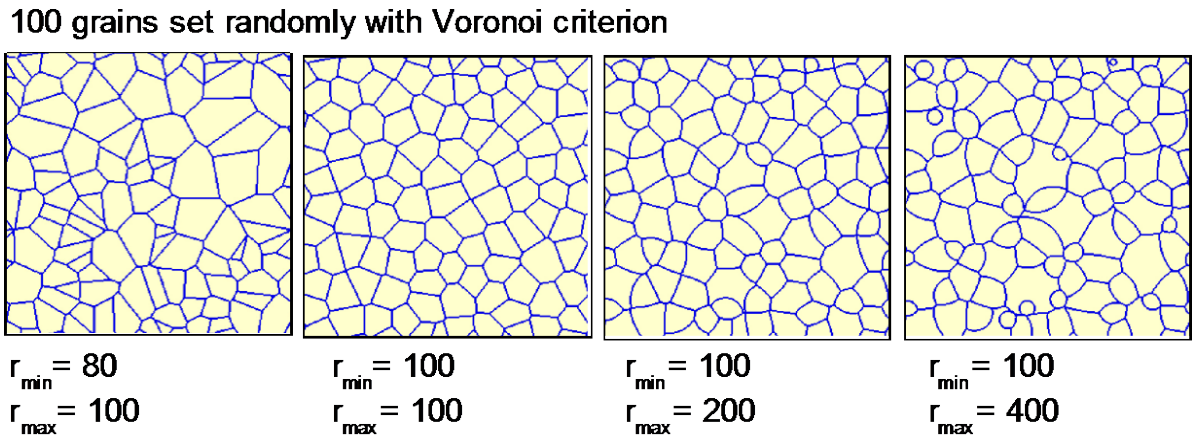


Figure 4.6 Applying the Voronoi criterion

4.7 Data for further nucleation

4.7.1 The special status of phase number 0

The phase number 0 has a special status in MICRESS® as it is pre-existing at the beginning in the whole simulation domain, defined as one single grain (with number 0) without any anisotropic properties. In solidification simulations this is typically the melt which is anisotropic and which can form only one “grain“. Therefore, for simulation of nucleation from the melt, no initial grain needs to be defined at all.

The predefined grain 0 of phase 0 has the advantage that a microstructure never can have “holes“. But for simulation of solid-solid reactions it may seem somewhat strange. For the gamma-alpha phase transformation, for example, phases 1 and 2 have to be used for the gamma and alpha phase because both consist of several grains and eventually have orientations. Typically, the initial grain structure of phase 1 is defined by setting randomly grains into the phase 0 background, and if their radii are too small, free areas of phase 0 could be left over. MICRESS® does not recognize this automatically as an error, because, in principle, this could have been done by the user with intention.

In case of the gamma-alpha transformation, the user should take care that there is no liquid phase left and it is not necessary to define any of the phase interactions for phase 0 with the other phases. When using the TQ interface, no identification of phase 0 in the database is necessary. If, however, latent heat is used, then for phase 0, even if not present and if no phase interactions are defined, thermophysical data (enthalpy and heat capacity)have to be specified by the user: for the case that phase 0 would exist, even as inert phase, the average heat capacity would depend on these thermophysical data!.

On the other hand, the user has the freedom to assign any phase number to the liquid phase, and in the case of Thermo-Calc coupling, even more than one (MICRESS-) phase can be assigned to the same thermodynamic phase (e.g. to distinguish primary and eutectic phases, delta and alpha ferrite, different liquids in monotectic reactions etc.).

Thus, the main inconvenience of the special status of phase 0 is that for most solid-solid purposes is remains unused but still allocates some computer memory, and in some cases a “dummy“ value for the enthalpy and heat capacity has to be given. Since MICRESS version 6.2 nucleation of phase 0 is supported using the option “add_to_grain” (see chapter 4.7.5).

4.7.2 Nucleation models

There are different nucleation models used by MICRESS® which refer to different types of phases or circumstances where nucleation is to occur. These are for example a specific interface, in the bulk or phase regions. Moreover, it is important to know whether a fixed critical undercooling or some other criteria based on the seeding particle properties, stored energies in recrystallisation, etc. shall be used as a nucleation criterion. The user speci-

fies which phase is nucleated on which substrate phase and in which phase the solutal undercooling of the nucleating phase is calculated (matrix phase). The user also determines the temperature range, the checking frequency, data for shielding subsequent nuclei, etc.

The following nucleation models are implemented in MICRESS®.

Seed density model

The seed density nucleation model implemented in MICRESS® [1] is based on a heterogeneous nucleation model similar to the one used by Lindsay Greer. A seed density-radius distribution function has to be specified which is defined stepwise (Figure 4.7). During heat extraction from the melt, the largest particles will nucleate first at an undercooling defined by their radius (if complete wetting is assumed or an effective radius is used instead). The particles start growing and releasing latent heat, while interacting with other potential nucleation precursors. Depending on the heat extraction rate and the amount and dimensions of all other seeding particles, the temperature will drop more or less below the liquidus temperature, thus defining the amount of seeds to be activated According to the model, the effectiveness of inoculants added for grain refinement is defined by:

- the maximum particle size
- the particle size distribution

In MICRESS®, the seed density model should be used together with latent heat [1] or with coupling to the one-dimensional temperature field (1d_temp) [2] in order to allow an interaction of the potential nucleation sites via release of latent heat, but sometimes it may seem appropriate to use it just as a simple way to randomly distribute a given number of nuclei in a defined region. The heat extraction rate dependency of the number and size primary silicon particles in an Al-Si alloy under typical casting conditions is demonstrated in Figure 4.8 which also shows the nucleation of faceted primary Si in an Alusil alloy using the seed density model.

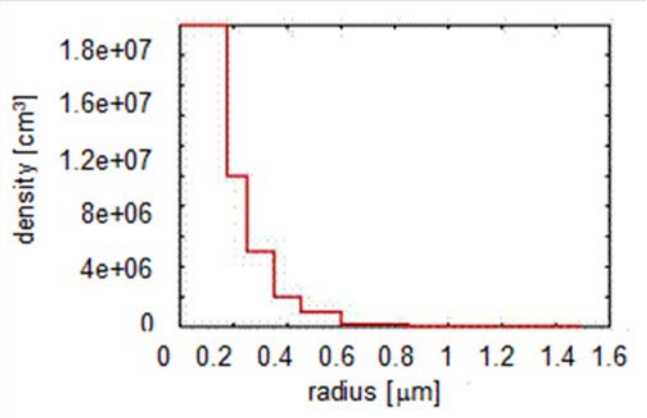


Figure 4.7 Seed density-radius distribution

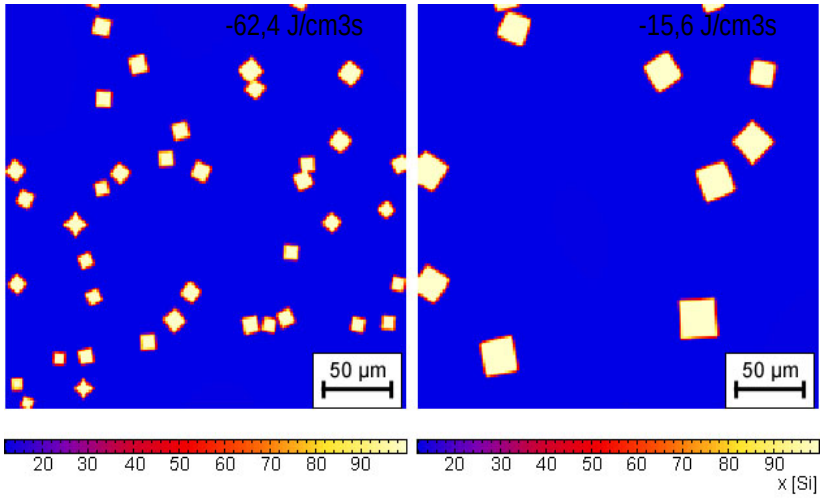


Figure 4.8 Nucleation of faceted primary Si in AlSi17Cu4Mg

Seed undercooling model

Using the seed undercooling model, a new seed is set if the local undercooling at a nucleant position exceeds a predefined critical nucleation undercooling. The local undercooling depends on the local composition and temperature [4].

In general, the MICRESS® nucleation models are designed for micro-scale simulations and not for the nano-scale, i.e. there is no model for the prediction of homogeneous nucleation based on thermal fluctuations for the critical seed formation.

4.7.3 Shield time and shield distance

The shield time and the shield distance parameters are related. They account for a physical shield effect due to the release latent heat or solute by a new grain which is not explicitly included in the simulation. This allows the user to set up a pragmatic nucleation scenario if the exact physical background is unknown. Supposed that at a time t a nucleus appears, the parameter "shield distance" will define a circular area with the diameter r_{shield} around the grain. No other grains of the same phase will be allowed to appear on this area within the time interval $(t, t+t_{shield})$. The r_{shield} parameter can be defined according to experimental micrographs of the system to be simulated and t_{shield} can be evaluated e.g. according to some typical diffusion time scale considerations. The optional parameter "nucleation distance" defines a radius $r_{nucleate}$ that replaces r_{shield} only for the time t of nucleation. Choosing $r_{nucleate} \geq 2 \cdot r_{shield}$ helps to distribute nucleations more evenly and over multiple time steps.

4.7.4 "Diffuse" phase model for unresolved eutectic or eutectoid regions

In many cases the length scale of eutectic microstructures is much smaller compared to grains or dendrites, and one may not be able to have both microstructure scales in one simulation. One typical example for that is the pearlite reaction which leads to the formation of lamellae in the submicron range while ferrite and austenite grains are orders of magnitude bigger. Treating pearlite as effective phase with its own phase properties is (at least in multicomponent systems) thermodynamically not consistent. Therefore we follow another approach which treats pearlite as a diffuse phase mixture of cementite and α-ferrite. The optional parameter "unresolved", which can be specified in the same line with the phase number for new nuclei, activates this model for the new grain and the corresponding parent grain which must be of the two eutectic phases. These two grains will not form an interface but mix as an extended triple-junction which is diffuse but can have locally different fractions. For further reference check the example "GammaAlphaPearliteTQ_dri" which is part of the MICRESS distribution.

4.7.5 Advanced grain manipulation using "add_to_grain"

This option which can also be activated in the same line with the phase number for new grains provides a bunch of possibilities for advanced "manipulation" of grains. The original intention why this option was included was the possibility to nucleate liquid phase during runtime. In order not to get unphysical "grain boundaries" inside the liquid phase, new liquid regions have to be formed by grain 0 which per default is already existing (at least in the data structure). Therefore, the task was generally to allow nucleation of grains which already exist, hence the name "add_to_grain". Having two or more not interconnected regions with the same associated grain number is nothing exceptional, so there is no reason why this should not be possible. The only side effects are that there will no interface be formed when these regions get in contact later (the same like for "categorized" grains, see Chapter 4.5.4), and that both regions are associated with the same grain properties like phase number, orientation, shield data, etc. But this latter fact is the reason why the new option provides several interesting possibilities for manipulating existing grains. But first let us have a look on the sub-options which are available with "add_to_grains" and which must be specified in the following line:

- "grain_number": The grain number of the new grain is explicitly specified. If this number already exists a second instance of this grain is created
- "parent_grain": The parent grain (the locally found grain with highest fraction of the substrate phase, or alternatively of the matrix phase if no substrate is defined) is used to define the grain number of the new

grain. As this grain already exists at this place, no change is done to the microstructure, but the grain properties of the parent grain are overwritten according to the nucleation parameters.

- "new_set": This seed type opens a new set of instances with the same but before unused grain number

With these three options, a number of smart applications are possible, e.g:

- Nucleation of liquid without formation of unphysical "liquid grains"
- Nucleation of a set of new precipitates with the same grain number ("new_set"). This reduces the total number of grains like the "categorize" option, but even more efficiently, and thus can greatly improve performance. Please note that due to their common identity, the "small grain model" may not work correctly (Chapter 4.6.1), and "fd_correction" (Chapter 3.2.3) should be used for compensation.
- A phase "switch": Using the sub-option "parent_grain", the phase number of an existing grain can be changed to another phase by using the nucleation mechanism. Martensitic transformations which occur instantaneously are a prominent example where this switch may apply.
- Removing rest of liquid in solidification is another very useful application of the "phase switch": Tiny amounts of remaining liquid often cause serious problems at the end of solidification, typically without being physically meaningful. Switching this remaining liquid (grain 0) to a solid phase is a very simple but effective method to avoid such problems!
- If the simulation of a chain of processes is performed, stored energy or dislocation density due to deformation processes can be applied during run-time or can be added to structures read using the "restart" functionality.
- Grain orientations can be continuously rotated during simulation by using "parent_grain" and "parent_relation".

Note:

- Not all manipulations which are possible are reasonable!
- Phase switches can cause numerical problems if the new phase has solubility restrictions!

4.8 Phase interaction data

When defining the phase interactions between two phases, the user may define either standard interactions or use some special models. E.g. in case of grain growth, the user may decide to use the particle pinning model, the solute drag model or recrystallization for the description of the grain boundary migration. Figure 4.9 shows the difference in the development of the average grain radius with time for the first two models mentioned above and normal grain growth.

The particle pinning model and the solute drag model will be described in more detail in the next sections.

Apart from these mesoscopic models, there are also advanced options for defining a dedicated redistribution behaviour of solute in the interface. Selecting this option allows the activation of the NPLE-model, the paraequilibrium model or an antitrapping current (ATC) for each element (see phase diagram input).

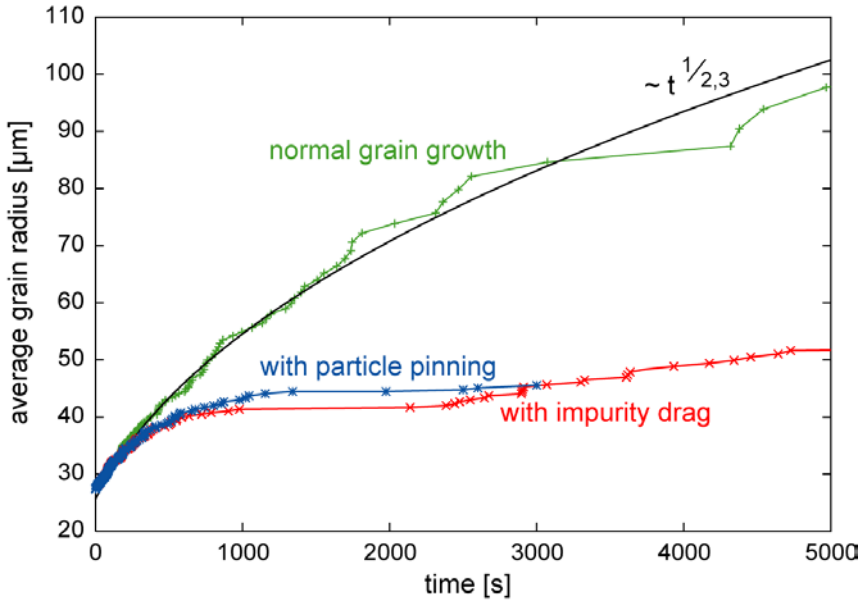


Figure 4.9 Normal grain growth vs. the particle pinning and the solute drag models

4.8.1 Particle pinning (Zener) model

The term “particle pinning“ describes the effect of small particles on the movement of grain boundaries. If the driving force for the movement of the grain boundary (typically the curvature) is not high enough to overcome the attractive interactions with the particles, the grain boundary is “pinned“ and will not move any more. Thus, the grain size may be reduced to a much lower value by the effect of particle pinning. In practice, particle pinning is an important mechanism in preventing softening of materials at high temperature due to grain growth. The particle pinning model implemented in MICRESS is a meso-scale model which takes into account the microscopic particle pinning effect without treating the individual particles explicitly. According to the Zener model, a critical pinning force $P^* \sim 3 \sigma c / r$ is defined, where σ is the particle surface energy, c is the “particle concentration“ and r - the particle radius (Figure 4.10).

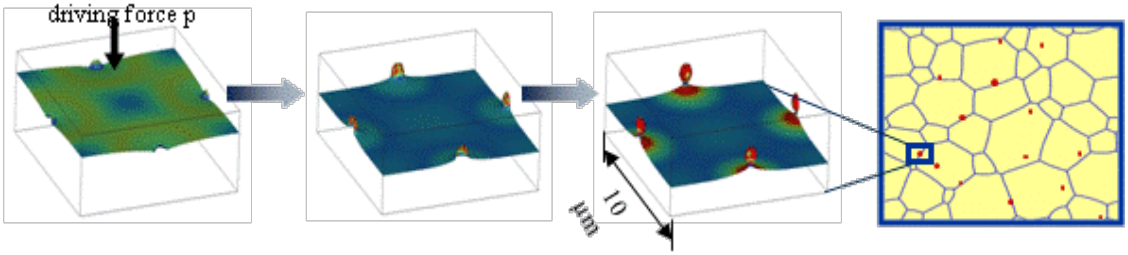


Figure 4.10 The particle pinning model

If the driving force p exceeds the critical driving force, the grain boundary will still move, but with a lower velocity. This velocity has been simulated for discrete particles using MICRESS® (Figure 4.10). The velocity curves can be described by an effective mobility which is implemented in the mesoscopic MICRESS particle pinning model. The user has to provide the critical pinning force p* in units of a critical pressure (MPa) (default option) or a critical curvature (1/µm) and a minimal mobility.

For grain growth, the driving force for grain boundary movement consists of the grain boundary curvature and the force balance at the triple junctions. In general, this driving force decreases with the average grain size. But especially the force balance at the triple junctions is not directly linked to the grain size. Therefore, with the particle pinning model, abnormal grain growth may be observed, e.g. one huge grain continues growing while the smaller

ones are already pinned.

From a theoretical point of view, the minimal mobility parameter should be zero. But in many cases, even if pinning occurs, still a very slow movement of the grain boundaries is observed, e.g. due to particle ripening. MICRESS gives the user the possibility to account for that by specifying a minimal mobility >0.

Figure 4.11 shows that using the particle pinning model instead of normal grain growth leads to different results. In the figure above, a clear tendency towards abnormal grain growth can be seen in the simulation where the particle pinning model has been used.

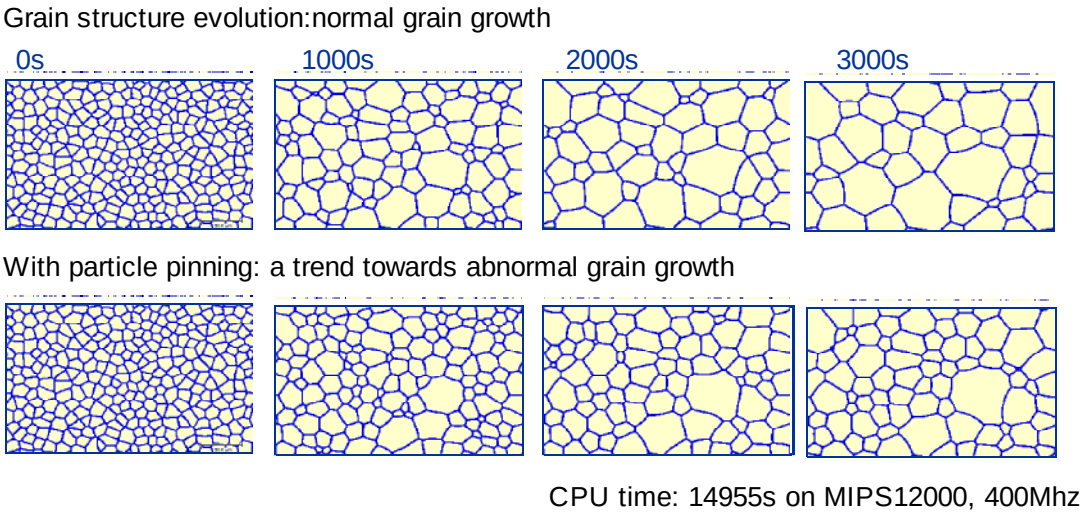


Figure 4.11 Grain growth without (top) and with (bottom) the particle pinning model

4.8.2 Solute drag model

The solute drag effect can be described as follows: in case of attractive interaction between the grain boundary and dissolved impurity atoms, the solute atoms will be concentrated at the boundary. When the boundary moves, the solute atoms will try to remain on the boundary, i.e. the boundary has to drag its impurity load and can only migrate as fast as its slowly moving impurities. If the actual velocity v is lower than the critical velocity v*, the solute atoms will stay with the interface, which is then referred to as a “loaded interface“ and is characterised by a decreased mobility. If the actual velocity v is larger than the critical velocity v*, the solute atoms cannot follow the movement and there is a “free interface“ formed, characterised by a high mobility. Thus, the impurity drag model uses a two step velocity dependent mobility of the grain boundary for description of the interaction of the grain boundaries with solute atoms (see Figure 4.12).

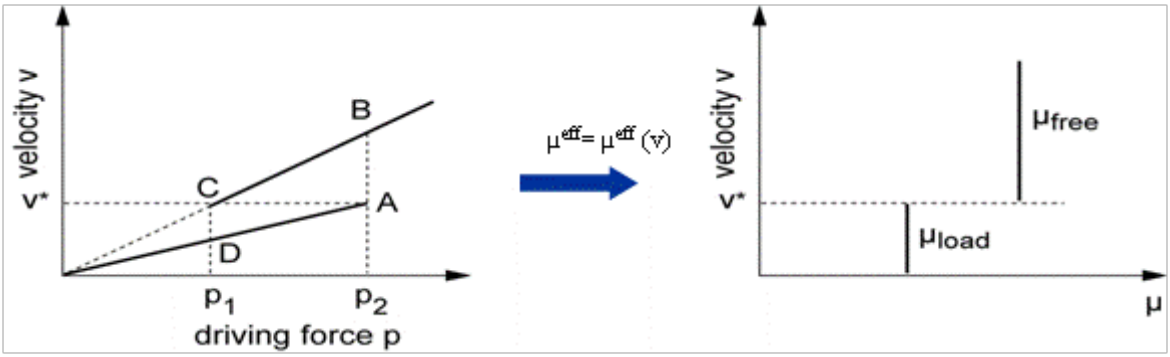


Figure 4.12 The solute drag model

Important parameters here are the critical velocity v* and the drag factor, by which the mobility of the grain boundary is reduced due to the drag effect. In order to prevent numerical fluctuations around the critical velocity, a transition range ΔV has to be specified: In the range between v* - 0.5 * ΔV and v* + 0.5 * ΔV, there is a linear transition between the reduced mobility μ* drag factor and the normal mobility μ.

4.8.3 Anisotropy of interfacial energies and mobilities

Interfacial properties can be defined 'anisotropic', i.e. as function of the interface normal vector $\vec{n} = (n_x,n_y,n_z)$ transformed into the coordinate system of the anisotropic grain. Two alternative model versions can be chosen to describe weak (non-faceted) anisotropy: The default 'standard' model and the new 'harmonic_expansion' model.

The standard anisotropy model is based on predefined anisotropy functions, listed in Table 4.1. The specific choice of the anisotropy function depends on the crystal symmetry which the user has defined for the anisotropic phase, as previously described in section 4.5.1.

Symmetry	Anisotropy function	2-d equivalent
Cubic	Stiffness: $a_{\sigma^*}=1-\delta_{\sigma^*}4(n_x^4+n_y^4+n_z^4)$	$a_{\sigma^*}=1-\delta_{\sigma^*}\cos 4\varphi$
	Mobility: $a_\mu=1+\delta_\mu 4(n_x^4+n_y^4+n_z^4)$	$a_\mu=1+\delta_\mu\cos 4\varphi$
Hexagonal	Stiffness: $a_{\sigma^*}= (1-\delta_{\sigma^*}(-n_x^6+15n_x^4n_z^2-15n_x^2n_z^4+n_z^6) -\lvert\delta_{\sigma^*}\rvert(n_y^6+5n_y^4-5n_y^2)) \cdot (1-n_y^2+n_y^2\cdot factor_y)$	$a_{\sigma^*}=1-\delta_{\sigma^*}\cos 6\varphi$
	Mobility: $a_\mu = (1+\delta_\mu(-n_x^6+15n_x^4n_z^2-15n_x^2n_z^4+n_z^6) +\lvert\delta_\mu\rvert (n_y^6+5n_y^4-5n_y^2)) \cdot (1-n_y^2+n_y^2\cdot factor_y)$	$a_\mu=1+\delta_\mu\cos 6\varphi$
Tetragonal	Stiffness: $a_{\sigma^*}= a_{\sigma^*}(cubic)\cdot(1-n_z^2+\cdot factor_z)$	z-elongated cubic
	Mobility: $a_\mu = a_\mu(cubic)\cdot(1-n_z^2+n_z^2\cdot factor_z)$	z-elongated cubic
Orthorhombic	Stiffness: $a_{\sigma^*}= a_{\sigma^*}(cubic)\cdot(1-n_z^2+n_z^2\cdot factor_z -n_y^2+n_y^2\cdot factor_y)$	z-elongated cubic
	Mobility: $a_\mu = a_\mu(cubic)\cdot(1-n_z^2+n_z^2\cdot factor_z -n_y^2-+n_y^2\cdot factor_y)$	z-elongated cubic

Table 4.3 Standard anisotropy function predefined in MICRESS®

The user can specify anisotropy coefficients δ_{σ^*} and δ_μ in the range of $\delta \in [0, 1]$. (A typical value for δ_μ in cubic metal systems equals 0.05.) Elongation factors can exceed the value of 1, but have to be positive. Table 3 illustrates the standard anisotropy functions for different choices of symmetry and coefficients.

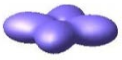
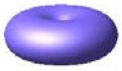
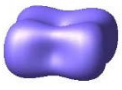
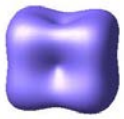
						
cubic δ=0.25	hexagonal δ=0.25, f_y=1	hexagonal δ=0.25, f_y=0	tetragonal δ=0.25, f_z=1.2	tetragonal δ=0.25, f_z=0	tetragonal δ=0, f_z=0	orthorhombic δ=0.25,f_y=.6,f_z=.8
						
cubic δ=-0.25	hexagonal δ=-0.25, f_y=1	hexagonal δ=-0.25, f_y=0	tetragonal δ=-0.25, f_z=1.2	tetragonal δ=-0.25, f_z=0	tetragonal δ=0, f_z=2	orthorhombic δ=-.25,f_y=.6,f_z=.8

Table 3: Examples of standard anisotropy functions with specific choice of coefficients

It is important to note that the static anisotropy is defined in the standard model with respect to the interfacial stiffness σ^* instead of with respect to the interfacial energy σ. The use of the the interfacial stiffness in the phase-field equation allows a direct matching to to the modified Gibbs-Thomson equation for anisotropic interfaces which reads: $v = \mu [\Delta G-\sigma^*\kappa]$ with v being the velocity in normal direction, μ the mobility, ΔG the driving force and κ the curvature, for more details see [Eiken 2009, thesis], page 120. In the 2-d case, the interfacial stiffness can easily be related to the interfacial energy by: $\sigma^* = \sigma + d^2\sigma/d\varphi^2$. However, this is not possible in 3-d, where curvatures in two different spatial directions have to be distinguished. Here, the stiffness model simply represents a pragmatic solution to consider a static anisotropy based on the same function as for the kinetic anisotropy. For a correct treatment of quantitative interfacial energy data in 3-d, the alternative anisotropy model which can be

selected by the optional keyword “harmonic_expansion” is recommended.

The harmonic-expansion model is especially helpful for 3-d simulations and also to incorporate quantitative interfacial data from experiments or Molecular Dynamics simulation. Here, the anisotropy functions are formulated by linear combination spherical harmonics selected in accordance with the crystal symmetry. The implemented harmonic functions are listed in Table 4.4 :

Symmetry	Anisotropy function	Spherical harmonics
Cubic	$a = 1 + \varepsilon_{c1}Y_{c1} + \varepsilon_{c2}Y_{c2}$	$Y_{c1} = 4(n_x^4+n_y^4+n_z^4) -3$, $Y_{c2} = n_x^6+n_y^6+n_z^6 -30\, n_x^2n_y^2n_z^2$
Hexagonal	$a = 1 + \varepsilon_{66}Y_{66} + \varepsilon_{20}Y_{20} + \varepsilon_{40}Y_{40} + \varepsilon_{60}Y_{60}$	$Y_{66} = \frac{\sqrt{6006}}{64\sqrt{\pi}}(-n_x^6+15n_{yx}^4n_z^2-15n_x^2n_z^4+n_z^6)$, $Y_{20} = \frac{\sqrt{5}}{4\sqrt{\pi}}(3n_y^2-1)$, $Y_{40} = \frac{3}{16\sqrt{\pi}}(35n_y^4-30n_y^2+3)$, $Y_{66} = \frac{\sqrt{13}}{32\sqrt{\pi}}(231n_y^6-315n_y^4+105n_y^2-5)$
Tetragonal	$a = 1 + \varepsilon_{20}Y_{20} + \varepsilon_{40}Y_{40}$	$Y_{20} = \frac{\sqrt{5}}{4\sqrt{\pi}}(3n_z^2-1)$, $Y_{40} = \frac{3}{16\sqrt{\pi}}(35n_z^4-30n_z^2+3)$
Orthorhombic	$a = 1 + \varepsilon_{20}Y_{20} + \varepsilon_{40}Y_{40}$	$Y_{20} = \frac{\sqrt{5}}{4\sqrt{\pi}}(3n_z^2-1)$, $Y_{40} = \frac{3}{16\sqrt{\pi}}(35n_z^4-30n_z^2+3)$

Table 4.4 Combinations of spherical harmonic function defined in MICRESS®

The contributions of individual harmonic functions are illustrated in Figure 1. The user can combine these contributions by the specific choice of the individual coefficients ε, which allows modeling of complex, manifold anisotropy functions. The choice of harmonic function for tetragonal and orthorhombic symmetry is only preliminary and shall still be extended according to future users demands.

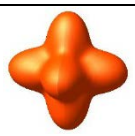
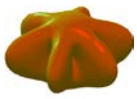
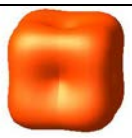
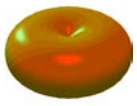
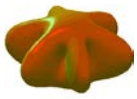
$1 + \varepsilon_{c1}Y_{c1}$	$1 + \varepsilon_{c1}Y_{c1}$	$1 + \varepsilon_{20}Y_{20}$	$1 + \varepsilon_{40}Y_{40}$	$1 + \varepsilon_{60}Y_{60}$	$1 + \varepsilon_{66}Y_{66}$
					
$1 - \varepsilon_{c1}Y_{c1}$	$1 - \varepsilon_{c1}Y_{c1}$	$1 - \varepsilon_{20}Y_{20}$	$1 - \varepsilon_{40}Y_{40}$	$1 - \varepsilon_{60}Y_{60}$	$1 - \varepsilon_{66}Y_{66}$
					

Table 4.5 Contribution of individual harmonic functions with positive and negative coefficient

In contrast to the ‘standard’ model, the static anisotropy is in the ‘harmonic_expansion’ model formulated with respect to the interfacial energy, which enters the free energy functional. The detailed derivation of the anisotropic phase-field equations is described in section 3.3 in [Eiken 2009,thesis].

4.9 Concentration data

4.9.1 Initial concentration

Manual input of a concentration-dependent diffusion coefficient

It is not possible to input a concentration- dependent diffusion coefficient "manually". It is assumed that the user has no information about concentration dependency, if no thermodynamic or mobility databases are used.

It should be mentioned that the main contribution typically comes not from the concentration dependency of the mobility, but of the thermodynamic factor. So, if there is no mobility database available for your system, but a

thermodynamic one, then an option would be to create a "dummy" mobility database with constant mobilities. Thus, the concentration dependency due to the thermodynamic factor would be included automatically.

Read initial concentration from file

Using the flag “from_file“, MICRESS is able to read the initial concentration fields either from an ASCII profile or from a 2/3-dimensional ASCII array file. The easiest way of creating an ASCII input file is writing ASCII outputs for single time steps with Display_MICRESS. Such a file can afterwards serve as a template for the real data. For each ASCII file the dimensions (AnzX, AnzZ: number of grid points in x and z direction) have to be given, the compositions are interpolated to the actual dimension of the simulation domain. If necessary, the data can be flipped or transposed. If a thermodynamic database is used, a composition for initialisation of the thermodynamic description has to be given additionally. From these data, an initial equilibrium is calculated, and the corresponding thermodynamic description is stored for all the interface regions. These data will be used until a relinearization is performed.

4.10 Phase diagram – input data

4.10.1 Using linearised phase diagrams

The input required for applying linearised phase diagrams includes the reference temperature at which the linearisation was performed, the entropy of transformation, the concentration of each component in the considered phases, and the corresponding slopes (see figure 4.13).

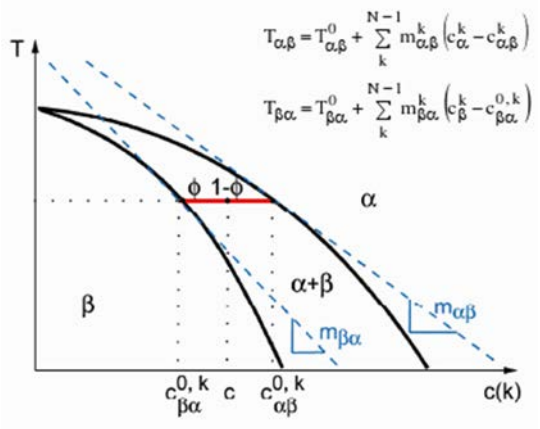


Figure 4.13 A linearised phase diagram

Use linearised phase diagram description if:

- no thermodynamic database is available
- the system is simple and quite linear (e.g. FeC)
- the considered process is (almost) isothermal
- you need to reduce the risk of numerical instabilities
- simulation time and memory usage are critical
- “thermodynamic noise“ is problematic
- MICRESS® is used for teaching purposes or qualitative modelling

4.10.2 Coupling to thermodynamic databases

TQ workspace overflow

If too many elements and phases are used with coupling to a thermodynamic database, it may happen that the data do not fit into the standard TC workspace of 100kB. In this case, the operating system may report "memory corruption" (or nothing) not always a useful warning is issued from side of the TC libraries.

In case of such a problem, the solution is very easy: one can specify the TC workspace size in kB as an extra line after the input of the GES5 file name.

Unfortunately, it is not easy to estimate the needed size beforehand, because it strongly depends on the complexity of the phase descriptions in the

Driving File

```
…
# Is a thermodynamic database to be used?
# Options: database no_database
database database_name
workspace_size 1000
…
```

Example 4.2 Adjusting of the TQ workspace size

database, which are extracted into the .GES5 file. There is also no strong correlation to the size of the .GES5 file. The default value of 100 kB was chosen such that no problems with most applications are expected but a performance loss due to this memory allocation is minimized. Up to now, to our knowledge, there was only one case using the ThermoTech database TTNI6 in connection with the γ' phase and a multicomponent alloy, where the default value was not sufficiently high.

Components with concentration 0

It is not possible to include a component with concentration 0 in a simulation with TQ-coupling, e.g. when more components than needed are defined in the .GES5 file. A composition of 0 is only allowed for stoichiometric phases, and then the corresponding composition needs to be specified for another phase which has solubility for this phase. This is not astonishing because a composition of 0 would correspond to an infinitely low entropy and thus an infinitely high driving force for dissolution of such a component from another phase.

A better procedure in such cases is either to use a .GES5 file without this element or is to specify a non-zero but sufficiently small value for this composition. But the user should be extremely careful, because this small value can nevertheless be important for numerical reasons and can affect performance!

When to choose coupling to thermodynamic databases?

Choose coupling to thermodynamic databases if:

- a full thermodynamic description including all phases and elements is available
- systems are complex (technical alloys, ternary or higher, more than two phases)
- a wide temperature range is addressed
- realistic behaviour is required
- you do not like to rack your brain about thermodynamics!
- Global vs. local thermodynamic description

While for local thermodynamic description (keyword “database“ or “database local“ in the definition of the phase diagram input per phase pair) each interface cell has its own thermodynamic description which is obtained from the database when the interface cell is newly formed or when the description is updated, with global thermodynamic description (keyword “database global“) only one such description is used for a complete interface between two grains. The local thermodynamic description is default and should be used as such because it is safer and more exact. At triple junctions, consistent descriptions for all pair-wise interactions are assured, because the same mixture composition in the considered grid cell and identical phase fractions are used while updating the data from database (it is also taken care that all interfaces in a triple or higher junction are updated even if only updating for one phase pair is requested.

If a global thermodynamic description is used for a certain phase pair, this consistency cannot be assured. But, as compensation, a mostly significant reduction of the calculation effort for this phase pair is achieved (which can be checked in the .TabTQ output). This often gives the possibility to afford using a smaller updating interval which may compensate for the loss of numerical stability. The use of global thermodynamic descriptions can be worthwhile for phase interactions between precipitations which are small and thus do not feel much differences in composition or temperature with their surroundings. This option may also be successfully used if numerical noise inside an interface which is connected to the phase diagram description is a problem, e.g. at the very end of a solidification simulation or during Ostwald ripening of particles of “thermodynamically difficult“ phases like MC carbides.

Which updating schemes should be chosen?

There are presently 5 different updating schemes for thermodynamic data available:

- Global updating interval
- Updating interval per phase pair
- Critical effective temperature deviation per phase pair (“automatic”)
- No explicit updating (“none”)

The most simple and reliable method is using a global updating scheme. By this way, a complete updating of all thermodynamic descriptions is assured and the computational effort can be controlled. The smaller the updating interval, the more accurate are the results but the higher is the computational effort. The user should use a value which is small enough not to affect considerably the results but high enough to fulfil his performance requirements.

Using updating per phase pair should be considered if certain phase pairs need special care, and the strong performance loss connected with a small global updating interval has to be avoided.

NOTE: If updating per interface and global updating is combined, extra effort can be created if the intervals are no exact multiples. Also, in case of pure updating per phase pair and local thermodynamic descriptions, an extra effort is to be expected for triple and higher junctions if no exact multiples are used for the update frequencies (consistency of triple junctions!).

A smart method is to update thermodynamic descriptions only when needed. i.e. when the local conditions change sufficiently. An effective temperature deviation can be calculated for each interface cell using the temperature and concentration change since the last updating of this grid cell. With the updating method relying on a critical temperature deviation, the complete interface is updated if for one interface cell this limit is exceeded (updating only this single cell would produce huge noise which could even destroy the interface!). The main disadvantages of this method are that it can be quite difficult to figure out a reasonable value for the critical temperature deviation and that the computational effort is very difficult to predict

No explicit updating does not necessarily mean that the simulation is incorrect: All newly formed interface cells are automatically “up to date“, and if an interface is moving fast enough, the thermodynamic description is implicitly updated. Depending on the type of simulation (e.g. steady state growth using the “moving frame“ option), no explicit updating may be needed

4.11 Parameters for latent heat and 1D temperature field

If latent heat is used in a MICRESS® simulation, there is an additional source of numerical instabilities, which is not covered by the automatic time stepping criteria. If the interface mobility of a growing phase is too high and the time step is too long, the latent heat release in one time step can be so high, that the driving force created from the corresponding temperature change is also high. This would cause an even bigger (opposite) release of heat in the next time step. This behaviour is difficult to predict, because it depends not only on the thermodynamics of the system, but also on the amount of interface in the calculation domain and on the diffusion coefficients.

Consequently, there is no systematic way to prevent that, other than reducing the time step manually if necessary. This can be done by a proper specification of the time input data, e.g.

```
…
automatic_limited
constant
1.E-7 1.E-5 0.95 0.95
…
```

The first two parameters are the minimal and maximal allowed time step, the third and last parameter are prefactors to the phase-field and the segregation criteria; a default value here is 0.9. Note that not all of the four parameters need to be specified. The second value in this case would be used to decrease manually the time step to a numerically stable value. The first parameter is typically used for other purposes (e.g. if in a single cell numerical problems would cause extremely low time step values, which would practically stop the simulation).

Before limiting the time step value, one should check whether the correct resolution and interface mobility values are used and whether the latent heat parameter is reasonable.

If latent heat is used in a 2D-MICRESS® simulation, a 3D correction may be necessary, depending on the symmetry of the simulation domain. If all three spatial directions are equivalent like in equiaxed solidification, the 2D shape corresponds to a projection of the 3D structure to the 2D plane. As a consequence, there is a systematic error of the 2D phase fraction of the primary phase, leading to a corresponding error for the release of latent heat. By use of the keyword "lat_heat_3d", a correction may be possible, depending on the situation. There are two different situations:

- If, e.g. for a growing equiaxed dendrite, the solutal diffusion fields do not touch the calculation domain boundaries (initial stage of growth), then growth in the third direction (perpendicular to the 2D plane) can be considered as independent of growth in the 2D plane. In this case, a geometric correction of the fraction of the primary phase is possible:

$$f^*_\alpha=(f_\alpha/(f_\alpha+f_{matrix}))3/2*(f_\alpha+f_{matrix})$$

  Here, f_α is the fraction of a growing phase, and f_{matrix} is the fraction of the (vanishing) matrix phase.

- If the solutal diffusion fields already touch the boundaries of the simulation domain or diffusion fields of other dendrites, growth in the three dimensions is no longer independent, and mass balance has to be considered. In this case, the 2D fractions are correct and need no transformation.

Thus, one would need a 3D correction in the initial stage of solidification (e.g. for heterogeneous nucleation using the seed density model, where these corrections are essential!), but switch it off at a later stage. The critical fraction is used for this purpose

$$f^{**}_\alpha =(f_\alpha/(f_\alpha+f_{matrix}))^{3/2}*(f_\alpha+f_{matrix}^*)$$

$$\text{with:}\quad f_{matrix}^* = f_{matrix} - f_{crit} \text{ if } f_{matrix} > f_{crit} \quad \text{else} \quad f_{matrix}^* = f_{matrix}$$

f_{crit} is the critical fraction of the matrix phase at which the solutal diffusion fields substantially touch each other or the domain boundary.

This part of the matrix phase is regarded as an inert phase with respect to the correction of the phase fractions. Therefore, it is subtracted from the matrix phase fraction. When f_{matrix} comes close to f_{crit}, the correction is automatically switched off smoothly. Unfortunately, f_{crit}, has to be estimated by the user, which results in an additional “fiddle” parameter.

4.12 Boundary conditions

4.12.1 Pressure condition for TQ

When using a database the pressure condition for thermodynamic calculations defaults to the standard of 101325 Pa. To change this parameter switch on TQ volume control (see section 3.2.14) and enter the pressure at the start of the boundary conditions. One use is to set the pressure for a system including an oxygen gas phase to the partial pressure of oxygen in air. This pressure is completely independent of any pressures calculated in the flow module.

4.12.2 1d far field approximation

The 1D far field approximation is particularly useful for decoupling phenomena occurring on different length

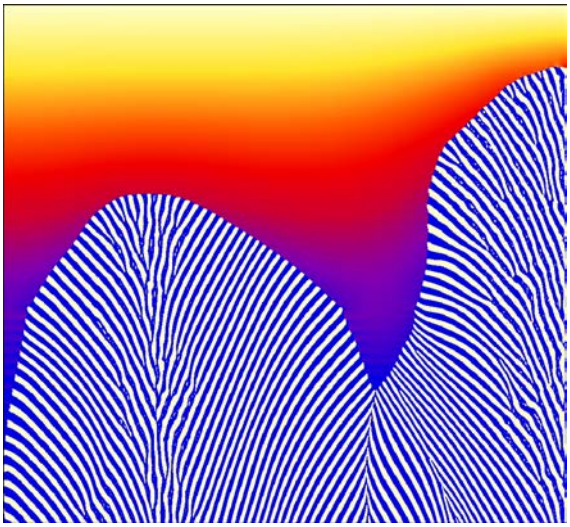


Figure 4.14 Solidification of Al-5.0 at% Ag-24.5 at% Cu using the 1d far field approximation V = 2.44 mm/s, G = 27 K/mm

scales like e.g. the complex diffusion in the vicinity of a eutectic solidification front in contrast to a one dimensional concentration field far away from the front, Figure 4.14. This section will be further elaborated in the next edition of this manual.

4.12.3 1d temperature field

If the temperature field is neither linear, nor isothermal, temperature diffusion is much faster than solute diffusion. That is why the temperature must be solved on a coarser grid. However, there is usually a temperature gradient present, i.e. a preferred direction for which the temperature field can be approximated as a 1D-problem.

Coupling to a 1d temperature field is reasonable especially in cases where instationary temperature fields are present (e.g. continuous casting, see Figure 4.15). Coupling is also useful in case of non-linear temperature fields as well as for cases where a temperature gradient is present, but the latent heat is also a matter of interest.

Important: the initial position of the bottom of the 1d temperature field is always negative with respect to the micro-domain (Figure 4.16).

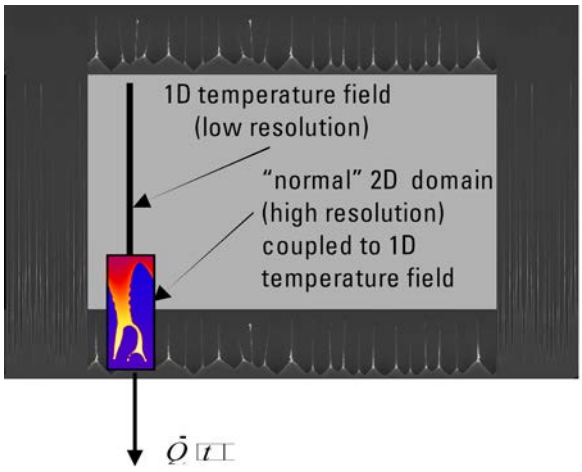


Figure 4.15 1D temperature field

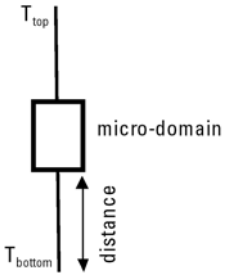


Figure 4.16 1d temperature field with respect to the micro-domain

4.12.4 Thermal boundary conditions

The thermal boundary conditions are specified according to whether or not latent heat is considered. The two cases will be treated separately.

no release of latent heat:

Latent heat can be disregarded in a simulation process if a strong temperature gradient is present (e.g. Bridgman process), if the process is temperature-controlled or if the transformation enthalpies are small. In these cases temperature curves can be specified by either starting temperature and cooling rate or as complex temperature curves (Figure 4.17). An example of a simulation with no latent heat release and a temperature gradient is the peritectic dendritic formation in Fe-C-Mn (see Figure 4.18).

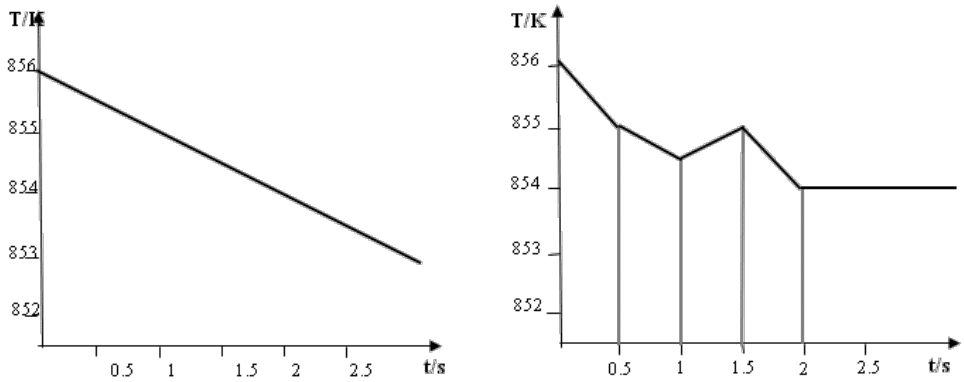


Figure 4.17 Time temperature curves specified by a starting temperature and a cooling rate (left) and as complex curves (right)

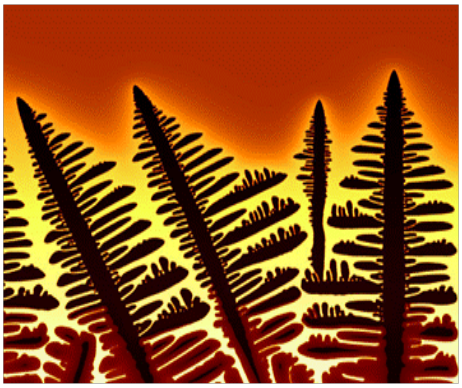


Figure 4.18 Peritectic dendrites in Fe-C-Mn

release of latent heat:

It is important, if microstructure evolution affects the temperature field. The temperature is calculated by balancing latent heat with net heat extraction. It should be noted that it is not compatible with the moving-frame option. In this case, the heat flux over time shall be specified (see Figure 4.19).

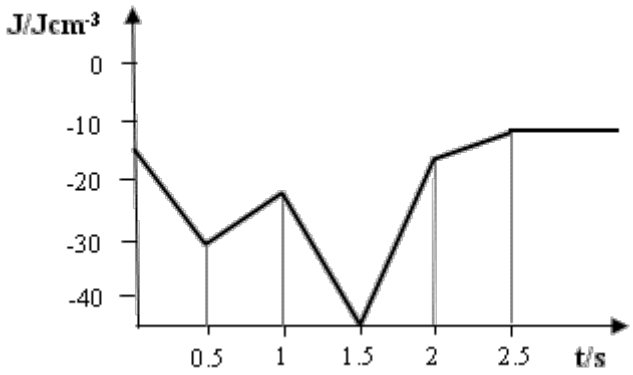


Figure 4.19 Defining heat flux over time for latent heat release

Figure 4.20 shows a model of an automotive piston where latent heat release has been considered. The material here is a five component aluminium alloy. In the numerical modelling, the homogeneous melt is cooled with a constant heat extraction rate. A seed density model has been applied for nucleation of primary silicon particles.

During growth of the primary silicon, the melt is depleted from silicon [14].

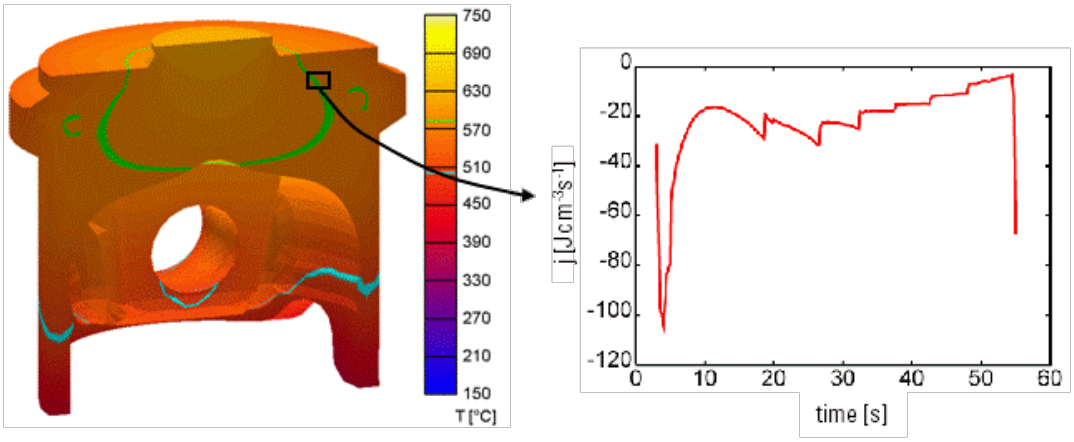


Figure 4.20 An example for latent heat release: a five component Al-alloy

4.12.5 Periodic boundary conditions

Using periodic boundary conditions with experimental microstructures is a fundamental problem, because experimental structures are never periodic. In that case, an extra grain boundary is formed all along the domain boundary (see the Grain-Growth examples on the MICRESS® distribution CD). At the beginning, what happens may seem quite strange, but with proceeding grain growth, the result becomes acceptable.

4.12.6 Gradient boundary conditions

Gradient boundary conditions should be used if you don't want to obtain additional interfaces. Both isolating and symmetric boundary conditions lead to interfaces which are perpendicular to the border of the domain and this is bad for grain growth.

IMPORTANT: gradient BC should be used ONLY for the phase-field. Using them for the concentration field would not only make no sense from physical point of view, but would also lead to instabilities if interfaces between different phases touch the calculation domain boundary.

4.12.7 Cooling rate and temperature gradient

In the standard case, one can assume a linear temperature profile on the microscopic scale. This is the reason, why a cooling rate and a temperature gradient are specified. The meaning of this assumption is that the temperature gradient at the dendrite tip cannot change; only the temperature can change. The relation between the temperature at the tip and the one at the bottom (considered by the input file) depends only on the distance bottom-tip and the given temperature gradient. For a more precise approximation, the “1d_temp“ option can be used. It applies an external one-dimensional temperature field. In this case, however, thermal boundary conditions need to be specified.

4.13 Numerical parameters

4.13.1 2D simulations of 3D structures?

Very often memory usage turns into a problem for the performance of 3D simulations which are expensive concerning both time consumption and memory usage. In such cases it is often reasonable to perform 2D simulations of the given 3D structures. But how to choose the correct 2D parameters in order to obtain a structure that resembles the original 3D one the best way? How close are the simulation results of one and the same structure, once simulated as 2D and once as 3D? The following cases allow you to perform a 2D simulation:

- No curvature in 1D e.g. planar solidification front and lamellar eutectic structure: 2D simulation is correct
- One axis is predominant e.g. due to a temperature gradient: 2D simulation deviates from the 3D one within a small factor in the order of $\sqrt{2}$ (See Figure 4.21.)
- No axis is predominant (see Figure 4.22)

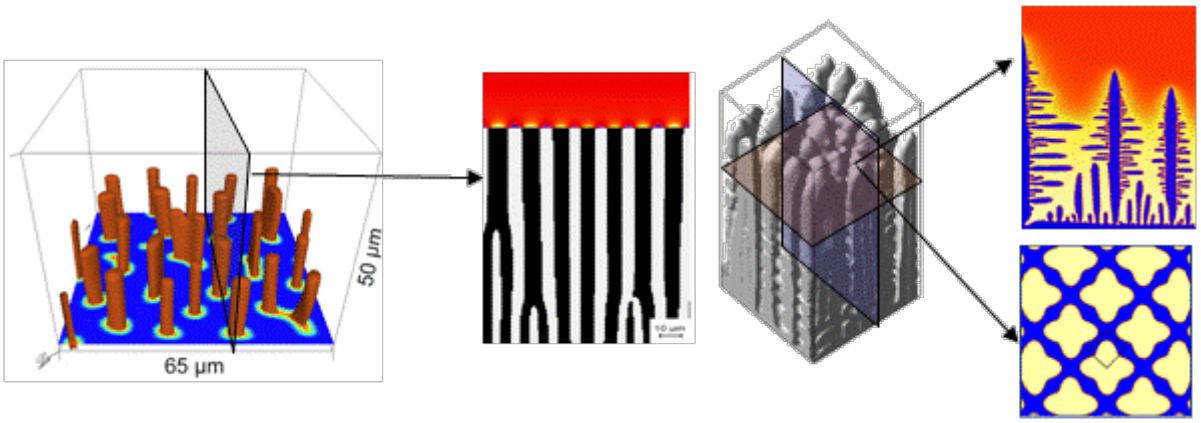


Figure 4.21 One axis is predominant

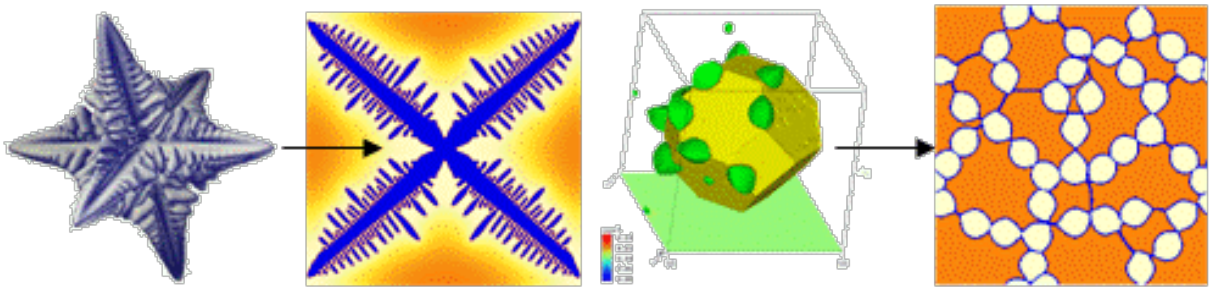


Figure 4.22 No axis is predominant

4.13.2 Grid resolution and interface mobilities

The number of grid cells, i.e. the resolution for a given domain size, strongly affects the simulation time. Assuming that diffusion is the performance bottleneck, one can expect a factor of 16 in simulation time, if the value of ΔX is divided by 2! Therefore it is very important to find the optimal resolution.

In general, the resolution is high enough, if a small change in ΔX does not change the results. In order to make a systematic study it is important to reduce the problem (temporarily) to the smallest representative size (taking advantage of all possible symmetries). With this small domain size it should be possible to get down to the really high resolutions where the results do no longer depend on the resolution. For this test simulation, all critical physical parameters (diffusion coefficients, interfacial energies, interface mobility) must be already adjusted because

they would change the test results. To quantify the differences, one can choose e.g. the phase fraction of the growing phase or the temperature, if latent heat is used.

The procedure would be the following: starting from the highly resolved test simulation, one would increase stepwise ΔX and compare the chosen quantity vs. time (e.g. phase fraction), observing from which ΔX the Δt_s gets substantially different. Then, one would go one step back and use the corresponding ΔX for the big simulation.

Unfortunately, in many cases one would still be in a ΔX range where performance is poor. In that case, one could go one step further: typically, a too low resolution for a diffusion-limited phase transformation leads to an artificial solute trapping and increased front kinetics. To a certain degree, one can try to compensate for that by reducing the interface mobility. As long as the development of the phase fraction can be corrected in this way, and the morphology is also retained (a different morphology earlier or later will also affect the fraction-time curve!), there is no reason not to do that!

If one is willing to invest even more calibration effort, one can use additional numerical parameters like the DG-options (averaging length etc.) to even better adjust to the correct interface kinetics and phase morphologies.

Related to that is also the choice of a reasonable value of the interface thickness. If performance is critical, one can go down to 3.5 cells without deteriorating the curvature evolution too much.

Of course, all these measures may decrease the exactness of the microstructure predictions for a given domain size. But, if due to the better performance a larger domain can be simulated, the exactness of the overall result may be increased!

The proper choice and meaningful adjustment of the numerical parameters are crucial for the successful numerical prediction of the results. The most critical parameters for numerical stability are:

- grid resolution: must be high enough to resolve the diffusion profiles, related to the diffusion coefficients and the growth velocity of the surface and the curvatures of the finest microstructure, expected from it;
- interface mobility: must be high enough to allow diffusion-controlled growth without kinetically slowing down the interface.

This means that using very high resolution and high interface mobilities would automatically result in stable simulations. Indeed this is true in most cases, but it is not practical because it results in extremely high calculation times! For example, increasing grid resolution by a factor of 2 in a 2-dimensional simulation will result in a factor of 4 (8 for 3D) for the simulation time, if the simulation is limited by interface related load (high proportion of interface in the calculation domain), or by a factor of 16 (32 for 3D), if diffusion is time-limited! Increasing the interface mobility proportionally reduces the phase-field time step and thus increases the simulation time.

As a general approach for finding proper numerical parameters:

4. Try to use the smallest size of the simulation area which still represents your problem. After finding suitable parameters you can apply them to larger domains or more complex problems.
5. Reduce the problem to only one interface between two phases. Optimising different phase interaction at a time will produce confusion!
6. Use the automatic time stepping option to get rid of the time-step as an additional numerical parameter.
7. Don't use physical parameters for optimising numerical stability! If you do not know all of them, make guesses and stick to them, until later you can calibrate them by comparison of stable simulations to experimental data.

Depending on whether one needs quantitative or just qualitative results, and whether one can afford the high resolution, the following three empirical approaches can be used:

Approach 1: Qualitative work

Choose a resolution for which you have the feeling that you can afford (in terms of calculation time). Start with a guessed mobility value. If the simulation crashes or shows instabilities, lower the mobility value until you get a

stable run. Look at the ".driv"-output to see whether the movement of the interface is slowed down strongly by the choice of your mobility! If that is the case, the grid resolution is not fine enough. Try again with a smaller value. On the other hand, if the simulation runs properly with your first guess of the mobility and resolution value, try to reduce the mobility or increase the grid spacing to make your simulation even faster.

Approach 2: Full resolution of the microstructure and diffusion profiles

If you want to be sure that the choice of the grid resolution and interface mobility does not affect the results of your simulation, you need to check quantitatively the growth velocity of the interface. This can be easily done by plotting one of the phase fractions vs. time (.TabF output) or the temperature vs. time (.TabL), if release of latent heat used. Then you can check quantitatively the growth kinetics for different parameter settings. The way to proceed is the following:

- Start from a given grid resolution and modify systematically the interface mobility. You will experience that growth kinetics will get faster with increasing the interface mobility, as long as you are in the range of too low mobilities. If you further increase the interface mobility depending on the grid spacing, you will get the following behaviour:
- grid spacing is too large: The growth rate of the interface will increase further until you end up in numerical instabilities.
- grid spacing is small enough: The growth rate will reach a plateau and remain constant even for higher Interface mobility values.
- Once you observe such a plateau, you can be sure that (at least from numerical reasons and assuming diffusion-limited growth!) your results are quantitative. Unfortunately, there are only few cases where you can afford to solve your problem completely with that high resolution. In most cases you have to proceed to Approach 3.

Approach 3: Quantitative without fully resolving microstructure and diffusion profiles

A way to get the right kinetics even if you cannot afford the high resolution you would need for complete resolving the microstructure and diffusion profiles is to use the interface mobility to calibrate the growth kinetics. This necessarily means that you need to know the exact kinetics of your process either from experiments or by getting it from a high resolution simulation via Approach 2, using the smallest possible calculation domain which still represents your simulation problem.

Calibration is just done by variation of the interface mobility for the given (too large) value of the grid spacing and comparing (via .TabL or .TabF output) with the reference simulation (or experiment). This is not always possible if the morphology of the low-resolution simulation is very different to the real one. In that case, you should try to use the averaging of the driving force across the interface to match better the morphology. You can adjust the averaging length of the driving force separately for each phase interaction. In this way, in many cases you will be able to perform low-resolution simulations without getting all microstructure details but with the appropriate growth kinetics.

As shown, it is quite a bit of work to perform quantitative simulations with MICRESS®, if there is no access to infinite computational power or if the problem is not extremely simple. But keep in mind that in many cases you may be just interested in knowing how a system reacts on changes of a physical parameter, or due to the lack of physical parameters you anyway need to calibrate your simulation to experimental results. Then, it is often not necessary to perform a perfect adjustment of the transformation kinetics to get a valuable insight into the problem by simulation.

4.13.3 Type of potential

The functions "double_well" (for MICRESS versions < 5.5), "double_obstacle" and "multi_obstacle" (for MICRESS version > 5.4) denote the type of potential which is assumed for the free energy description of the diffuse interface. These potentials lead to a bit different phase field equation, concerning the e.g. the polynomial part and the weight factor for the driving force.

- The double-well equation has the form:

  $\delta phi/\delta t \sim \sigma (\Delta\phi + 36/\eta^{**}2\ \phi\ (1-\phi)\ (\phi-1)) + 30/\eta\ \phi^{**}2(1-\phi)^{**}2\ \Delta G$

- The equation for double-obstacle has the form:

  $\delta phi/\delta t \sim \sigma (\Delta\phi + \pi^{**}2/\eta^{**}2\ (\phi-1/2)) + \pi\ /\eta\ \ sqrt(\phi(1-\phi))\ \Delta G$

The “multi_obstacle“ is based on the double_obstacle and additionally comprises higher order junction terms allowing for a correct description of wetting phenomena and triple junctions [16].

The double-well potential yields a hyperbolic-tangent profile as a steady state-solution for the interface contours. The solution obtained by the double-obstacle potential is half a period sine (or cosine) profile.

Both contours are quite similar in the middle of the interface (same gradients), but differ at the outer parts. The sine-function is steeper and gives a finite interface thickness (controlled by the input parameter "interface thickness"), the hyperbolic tangent has no finite thickness!

However, to keep the multiphase/grain problem numerically feasible, the tanh-profile is cut at ϕ = "phase minimum". Nevertheless, the interface is twice as thick as the specified value for the same "interface thickness".

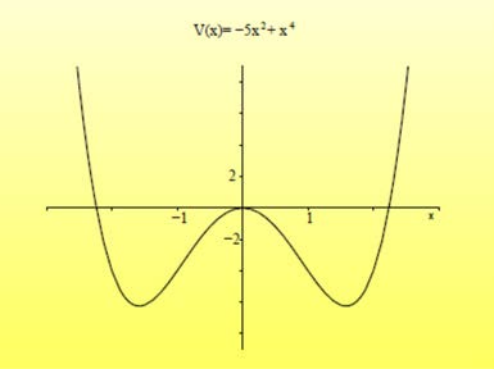


Figure 4.23 A plot of the double-well potential

The double well potential - having its origin in classical Ginzburg Landau theory - was removed from standard MICRESS versions in view of the advantages of the double obstacle formulation.

4.13.4 iface and nTupel

In most cases, the iFace and nTupel parameter values are adjusted automatically. Under special conditions one might wish to be able to control the field sizes in a better way. This may be in the following cases:

- the user is interested in having a static list size. Then, the second optional parameter should be 0.
- the calculation begins with a lot of grains (e.g. grain ripening). Starting with 0 as initial size would lead to a high number of increasing list actions during initialisation. This could take some extra time!
- the user is interested in keeping the list size as small as possible. In that case, the allocated size can be adjusted to the actual size. This can be achieved by a proper specification of the second and third parameter. Example 4.4 means for iFace that if less than 80 % of the allocated iFace-Field is used, it will be reduced so that 90 % of it is used after the reallocation. On the other hand, if in the case of a growing iFace list 100 % usage is reached, then

Driving File

```
…
# Phase field data structure
# ----------------------------------
# Coefficient for dimension of field iFace
1.0  0
# Coefficient for dimension of field nTupel
0.5  0
…
```

Example 4.3 List size optimisations

Driving File

```
…
# Phase field data structure
# -------------------------
# Coefficient for dimension of field iFace
1.0 0.8 0.9
# Coefficient for dimension of field nTupel
0.5 0.7 0.85
…
```

Example 4.4 List size as small as possible

the iFace field is increased so that afterwards 90 % are used. The default values are 0.25 and 0.75 respectively (iFace and nTupel).

4.13.5 Phase minimum phMin

The phase minimum (phMin, at the very end of the input file) is a purely numerical parameter which defines the fraction (phase-field parameter) below which an interface cell is assumed to convert to a bulk cell. If the fraction of a phase is below this value, the concentration of the small rest amount is frozen into the other phase(s) which is (are) present in this cell. Consequently, high values of the phase minimum increases artificial solute trapping.

Small values of phMin reduce this effect, but may require smaller phase-field time steps for not producing instabilities related to segregation, and thus slow down the simulation. This is included automatically into the automatic time step function.

However, if a new phase appears in a grid cell, it has at least the fraction corresponding to the value of the phase minimum. In cases where the new phase has a stoichiometric composition much higher than the actual composition in the matrix phase, there may be not enough solute of this component contained in the grid cell, e.g. if the value of phMin is too high.

In most cases, a value of 1.E-4 can be recommended. Lower values down to 1.E-12 can be used if stoichiometric components with extreme concentration differences are involved (like precipitation of carbides from an extremely low carbon matrix).

4.14 Modelling solidification

4.14.1 Theoretical background on solidification

The solidification of metals occurs by nucleation of grains in an undercooled melt and their growth. One differentiates between homogeneous and heterogeneous nucleation. In heterogeneous nucleation, the nuclei formation is aided by the presence of impurity particles or mould surfaces. It is typical for the metals of commercial importance and requires an undercooling of only a few degrees [10]. Homogeneous nucleation will not be considered here.

There are three different growth mechanisms described in the literature: 2D nucleation growth, spiral growth and rough growth [7]. According to the 2D nucleation growth mechanism, the energy gain as a result of the formation of a (circular) nucleus is compared with the increase in surface energy. The latter is a result of the edge free energy from the critical nucleus radius. This is the growth model used by MICRESS®. The spiral growth and the rough growth mechanisms will not be considered here. For more information the reader shall refer to [7].

One of the major differences in the solidification of different metals is related to the nature of the solid-liquid interface. There are various mechanisms that describe the nature of the interface, the solidification process and the growth related to it on different scales (microscopic and macroscopic).The following brief review of the various mechanisms is based on the explanations on solidification found in [10]:

- microscopic models (atomistic mechanisms). Here, thermal, mechanical or chemical phenomena are studied at an atomic level. Two main models describe the solid-liquid interface as either diffuse or smooth.
  In the diffuse interface, the change from liquid to solid is assumed to occur over a number of atomic layers in which the liquid structure changes gradually to that of the solid. The model implies that the surface has a very high accommodation factor for liquid atoms. The accommodation factor describes the chance of a given atom (liquid or solid) to find a position on the other side of the boundary where its can attach itself. Thus, growth occurs continuously by addition of atoms at every atomic site (the so-called normal or continuous growth). Moreover, the motion from liquid to solid depends on the indices of the particular crystalline plane that faces the liquid. The less close-packed it is, the easier gets for the liquid atoms to attach themselves to the crystal. As a result, for a given amount of undercooling there is a difference on the growth velocity of the two crystallographic planes: the least close-packed grows with a greater velocity. However, the crystal shows tendency to assume faces that are close-packed or slow growing. This can be explained by the fact that the low-density planes of fast growth grow themselves out of existence leaving behind only close-packed surfaces.
  Atomically flat interfaces are believed to grow primary by the movement of ledges over the surfaces. When a screw dislocation emerges at the crystal surface, a ledge is automatically produced. A smooth interface remains stationary except during the passage of ledges. Ledges are also spread parallel to the interface. The so called lateral growth depends on whether a single nucleus forms per layer or many nuclei form simultaneously. Lateral growth is typical for many metals (e.g. Bi, Ga, Si) and intermetallic compounds. It shall not be considered further here.

- macroscopic models. Here, the analysed system is large enough and consists of large numbers of atoms. When the liquid-solid interface moves into an undercooled liquid whose temperature decreases in advance of the interface, the interface becomes unstable and in the presence of perturbations cells may grow out from the general interface into the liquid. The resulting dendrites may also become very complicated with secondary branches forming on the primary cell and tertiary branches forming on the secondary ones. Dendrite arms usually grow in thickness as they grow in length until they grow together to form a single nearly homogeneous crystal. In the absence of sufficient heat removal from the liquid, the temperature rises very rapidly to the equilibrium temperature because of the release of latent heat of fusion at the interface. The removal of latent heat of fusion from the interface is a very important phenomenon considered to have an effect on the nature of the solid-liquid interface on a macroscopic scale since it affects the rate of undercooling and thus the dendrite formation.

It should be mentioned that pure metals undergo dendritic freezing as a result from temperature undercooling, whilst alloys solidify as a result from constitutional undercooling. The latter results when a solid freezes with a composition different from that of the liquid from which it forms.

Surface tension is another macroscopic phenomenon that plays an important role on the nature of the solid-liquid interface. The effect of surface tension is to stabilise the phase-change boundary which is locally unstable due to the presence of dendrites. Without surface tension, undercooling would lead to a totally unstable system. The local instabilities (dendrites) are counter-balanced by the non-linear stabilising effects of the surface tension. Large surface tension in undercooling implies a negative deviation of the new equilibrium temperature from the melting temperature [5].

The main advantage of macroscopic models is that they allow for studying the gross behaviour of the matter and applying the laws of Newtonian mechanics. A more general approach than the macroscopic modelling is the one of statistical thermodynamics which ignores the detailed consideration of individual molecules and applies statistical methods to find the distribution of very large numbers of molecules [5]. An explanation of the “integer for randomization“ when generating initial microstructure in MICRESS® was already explained in chapter 3.2.10.

Intrinsically, solidification is considered to be an out-of-equilibrium phenomenon, since the undercooled melt does not solidify immediately. The amount of undercooling is generally small but can also be as large as 0.2 Tm (Tm is an ideal freezing temperature). A very pure metal has a fixed freezing temperature which is also its melting temperature. For alloys, the melting and the freezing temperatures are not the same and phase-change takes place over a certain temperature range. The phase-change from liquid to solid generally takes place at T<Tm. This is due to the effects of interface curvature and kinetic undercooling. The effect of the interface curvature describes the phenomenon of an increased “escape tendency“ of atoms at the surface of convex curvatures surrounded by less neighbouring atoms compared to atoms in the centre [5]. Kinetic undercooling is related to the decreased temperature at the interface and describes the amount by which this temperature differs from the equilibrium temperature [14]. Moreover, the temperature at regions closer to the interface boundary is higher than the one in the liquid far from the boundary. This phenomenon was already explained by the release of latent heat from the solidification front which causes the temperature increase. As already mentioned, the release of latent heat has a major role on the crystal growth mechanisms [4].

As shown above, dendritic growth is a common pattern of crystal growth observed during solidification of metals and alloys. Due to the fact that the dendrite morphology is directly related to the material properties such as hot cracking, corrosion resistance, yield strength, ductility, etc., an effective control of the solidification process is required.

Consequently, one of the big challenges for the successful prediction of microstructure is the understanding and modelling of dendritic growth. The aim of modelling dendritic crystal growth is to determine the profile of the solidification front and the growth velocity as a function of the undercooling, using the given material parameters: latent heat of fusion, specific heat capacity, equilibrium chemical potential, the change of chemical potential with changing concentration, diffusivity, bulk melting temperature, anisotropic surface tension ([11],[15]). The numerical modelling of solidification used here is based on the so-called Stefan problem of the moving interface in a diffusion field. The essence of the Stefan problem will be presented in the next section.

4.14.2 The Stefan problem of moving interfaces

The Stefan problem is a particular kind of a boundary value problem for partial differential equations adapted to the case in which the interface boundary can move with time [11]. It is generally used for heat transfer problems with phase changes in a pure material or alloy [8].

In mathematics, the phases are regions in which the coefficients of the underlying partial differential equations are continuous and differentiable up to the order of the corresponding equations. From physical point of view, these coefficients represent medium properties for each phase. The partial differential equations, however, are not valid at phase-change interfaces. To make them continuous, an additional condition is required – the so called Stefan condition. The Stefan condition defines the local velocity of a moving boundary (interface) as a function of some parameters. These parameters are evaluated at both sides of the interface and are usually derived from a certain

physical constraint. For example, the conservation of energy is the physical constraint in problems with heat transfer with phase change. If there are n > 2 distinct phases separated by (n-1) distinct phase-change boundaries, an n-phase Stefan problem can be defined. Some of its characteristic features are:

- uses the heat transfer phenomenon governed by parabolic equations
- any two distinct phases are separated by a sharp phase-change interface. The phase-change boundary moves, but there is an instantaneous equilibrium so that the laws of equilibrium thermodynamics can be applied.
- there is a known temperature at the phase-change boundary
- there is a discontinuous heat flux at the phase-change boundary and latent heat is released or absorbed at the interface [5]

Based on the written above, it may be said that the phase transition between liquid and solid in an undercooled liquid that leads to dendritic growth may be modelled using the Stefan problem. For alloy solidification, the undercooling of the melt is taken into account by redefining the equilibrium phase-change temperature with the help of the Gibbs-Thomson relation resp. the curvature effect and the kinetic condition. When both of these effects are considered together, the resulting equilibrium temperature relation is referred to as “the modified Gibbs-Thomson relation“. Besides the normal velocity of the interface, the modified Gibbs-Thomson relation includes an anisotropic mean curvature which corresponds to a certain surface energy. This surface energy depends on the direction of the interface normal. Moreover, the modified Gibbs-Thomson relation enables the dendritic growth by considering the influence of both anisotropic mobility and anisotropic interface on it ([5], [12]).

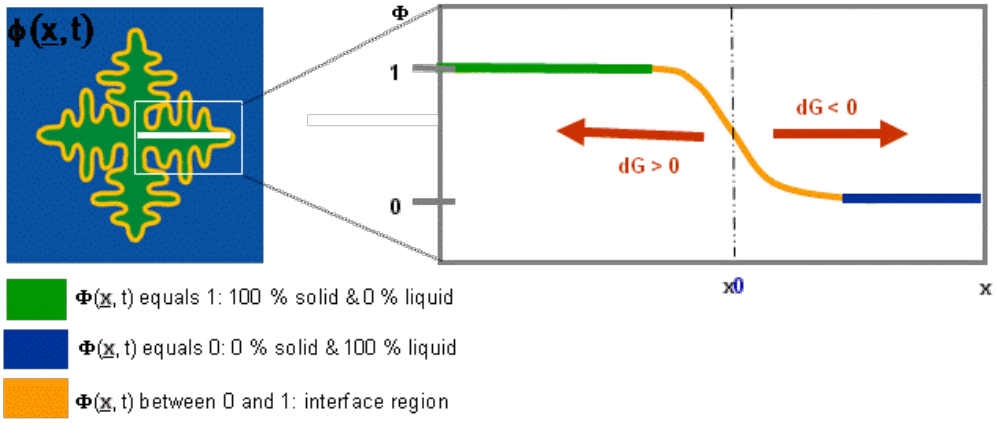
To sum up, MICRESS® uses the Stefan problem of moving interface for the description of solidification. Thus, sets of partial differential equations are defined that predict the profile of the solidification front and the growth velocity of the crystal. In order to solve these partial differential equations, the so-called phase-field method is used. It will be explained briefly in the next section.

4.14.3 The phase-field method

The phase-filed method is a method used to simulate microstructure evolution and changes in materials processing. It takes into account all boundary conditions at the liquid-solid interface, including anisotropy. Previous research has shown that morphologies computed with the phase-field method display most of the morphological features observed experimentally [9].

According to the phase-field theory, diffusion, ordering potential and the thermodynamic driving force are represented by differential equations of appropriate symmetry. The finite difference method is used to solve the corresponding differential equations. Its main characteristic feature is that the derivatives appearing in the differential equations are replaced by finite differences that approximate them [15].The solution of the differential equations describes the shape, the curvature and the motion of the diffuse interface. In equilibrium, for example, it yields a stationary interface profile with transition at x_0.

Depending on the conditions at the interface, there is a driving force dG which may lead to growth (dG < 0) or shrinkage (dG > 0). Characteristics of the phase-field model are (see Figure 4.24).



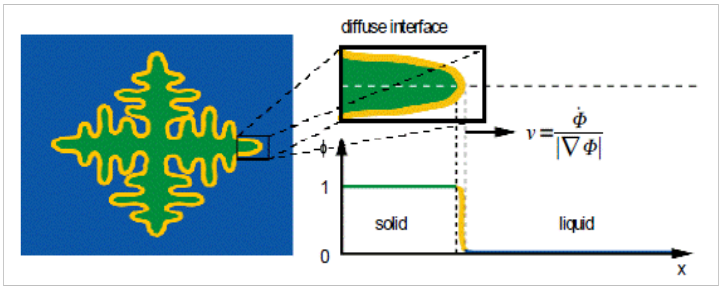


Figure 4.24 Main characteristic of the phase-field method: a diffuse interface

The main characteristics of the phase-field model can be summarised as follows:

The interface is described by a steep but continuous transition (in real space x) of the phase-field variable ϕ (x,t) between two states. The idea of a "diffuse interface" dates back to van der Waals. More information about his considerations can be found in [14]. The phase-field variable tracks the physical phase state at every point of the system, taking constant values in each of the bulk phases. It describes the distribution of either different phases or grains with different orientation. If the respective phase or grain exists locally, then its respective phase-field variable will obtain the value of 1, otherwise it will be 0 [1]. It varies rapidly between 0 and 1 in a region of thickness δ at the solid-liquid interface [15].

Non-equilibrium states are addressed in general. The phase field variable distinguishes between different states of a material that may be identical in all other state variables such as temperature, concentration, pressure, etc. Therefore, the phase-field variable is an independent state variable.

In the phase-field formulation, the Helmholtz energy F (ϕ, c, T) is a function of the phase-field variable ϕ_α. The Helmholtz energy is a result of the contribution of the energies of the bulk, the interface and the elastic part. The bulk phase consists of the bulk free energy densities but the phase fractions being given by the phase-field variables. The interface part is composed of all possible interface contributions between different phases or grains. Elastic contributions shall be added if there is significant volume expansion during phase transformation. The elastic contributions will not be considered here. The indexes α and β label grains which can have different crystal structures as well as grains which differ by their orientation. The index α can take any value larger than 1. Here, μ is the chemical potential, and c_α the composition of the phase field α. The symbol σ_α stands for the interface energy and $\eta_{\alpha\beta}$ – for the thickness of the interface. $\Phi_\alpha\Phi_\beta$ is the double-obstacle potential function [3]. The multiphase field

$$F = \int_\Omega f^{bulk} + f^{interface} + (f^{elastic})$$

$$f^{bulk} = \sum_{\alpha=1}^{N} \Phi_\alpha f_\alpha + \mu(c - \sum_{\alpha=1}^{N} \Phi_\alpha c)$$

$$f^{interface} = \sum_{\alpha=1}^{N} \sum_{\beta=\alpha-1}^{N} 4\sigma\alpha \frac{\beta}{\eta}(\Phi_\alpha\Phi_\beta - \eta^2\alpha\frac{\beta}{\pi^2}\nabla\Phi_\alpha\Phi_\beta$$

(MPF) model is to be used for an arbitrary number of different phases or grains of the same phase, but distinct by their orientation. According to the model, each grain α distinct from others either by its orientation or phase (or both) is attributed by its individual phase field Φ_α [14].

4.15 Using the flow solver

The flow solver simulates laminar fluid dynamics (Re $\lesssim 10^4$) for an incompressible liquid, and is applied to phase 0. All other phases are considered as immobile obstacles, that may be affected by phase field calculations but are not transported with the liquid. Advection is included in species transport for concentration coupled simulations. If concentration dependent density changes are defined buoyancy affects liquid flow by application of the Boussinesq approximation.

4.15.1 Activating the flow solver

The solver is activated with the [flow] or [flow_coarse] flag in the coupling section (see section 3.2.3). If [flow_coarse] is chosen an additional coarsening parameter *c* will be asked (this should be a low integer, preferably 2), and fluid flow will be simulated on a coarsened grid where each cell consists of c^d cells of the phasefield grid. The user needs to enter additional material data (see section 3.2.16), boundary conditions for velocity and pressure (see section 3.2.19 and section 4.15.3) and numerical parameters in the numerical section (3.2.20) discussed below (section 4.15.4).

Pressure and the velocity components averaged over the domain are printed at output timesteps. Summary warnings are provided if the convergence criteria are not met.

Restarts of fluid flow calculations will only work for the same cell dimension and coarsening factor of the simulation domain.

4.15.2 Composition driven convection

If any of the $d\rho/dc$ terms entered in the “Material Data for Fluid Flow” section is nonzero this feature will be activated. The density depends linearly on the composition and affects fluid flow according to the Boussinesq approximation.
To avoid numerical overhead, metallostatic pressures are neglected in the simulation. This means that buoyancy is evaluated relative to a reference composition. Usually this reference is the averaged composition of the liquid for the current timestep unless reference concentrations are given as optional parameter for each component in the material data section. While the averaging scheme works well if concentration changes affect the simulation domain more or less uniformly, a fixed reference works better for localized concentration changes, e.g. a dendrite growing from the bottom of the simulation domain. In this case the initial concentrations for the liquid phase are probably a good choice for reference concentrations.

4.15.3 Boundary conditions

The boundary conditions for fluid flow consist of a combination of velocity- and pressure boundary conditions. Basically three cases can be distinguished:

- Periodic, where the flow patterns of opposite boundaries are linked
- Conditions that fix the fluid flow across the boundary, i.e. fixed, symmetry, iso or parallel
- Conditions where pressure governs the flow across the boundary, i.e. orthogonal, free or gradient.

In the case of periodic velocity the pressure condition should also be “periodic”. The user will then be asked for a pressure offset. As shown in Figure 4.25 this adds an overall pressure gradient to the local pressure field.

In cases of fixed fluid flow across the boundary the pressure boundary condition does not affect fluid flow and can be chosen freely or set to “-“ (“no condition”). If all boundary flows are fixed outflow must match inflow.

If pressure governs the flow across boundaries (e.g. free in/outflow) at least one of the corresponding pressure

conditions should be “fixed” (e.g. on an outflow). For other boundaries (e.g. inlets) recommended pressure conditions are “fixed” or “von neumann”.

If the fluid flow is driven by a pressure gradient, friction must be present (either from solid phase in the simulation domain or from no-slip/fixed velocity walls parallel to the gradient) to avoid constant acceleration. Frictionless (or gradient) walls along the pressure gradient will also lead to situations with constant acceleration.

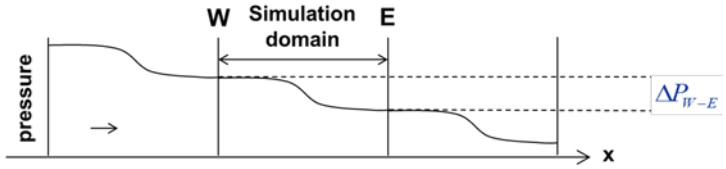


Figure 4.25: Pressure offset for periodic pressure conditions

- In the 2D regime the following velocity/pressure conditions will cover many cases:ppii/pp-- with some pressure offset and some solid phase in the simulation domain: simulation of permeability through a periodic array of grains/dendrites.
- ppfi/pp-- with zero velocity on the lower boundary (no-slip): poiseuille-like flow symmetric to the top boundary (parabolic velocity profile), e.g. for a dendrite growing perpendicular into a flow
- ppff/pp-- with zero periodic pressure offset, no-slip at the bottom and a fixed velocity parallel to the wall at the top: couette like flow (linear velocity profile)
- ffii/---- with the same, perpendicular velocity on the W and E walls: fixed volume flow

Iso boundary conditions can be replaced with symmetry conditions if that matches the geometry better. For pressure gradient driven flows without periodicity the pp../pp.. combination can be replaced with ee../ff.. or ee../nf.., and for the fixed volume flow ff../--.. can be replaced with fe../-f.. .

4.15.4 Numerical parameters

First the user has to choose a **solver scheme**. Of the three choices “combined”, “separated” and “piso” **in most cases “combined” is the best choice** for fluid flows through microstructures.

In cases of large velocities and large lengthscales the “piso” solver should be used. A good criterion to decide which solver to use is the Reynolds number Re:

$$Re = \frac{L \cdot v_{\text{avg}}}{\nu}$$

where L is the characteristic length (e.g. grainsize, DAS or domainsize), v_{avg} the average velocity and ν the kinematic viscosity.

For example L = 100µm, ν = 10^{-2}cm^2/s and v_{avg} = 10^4µm/s (=1cm/s) would result in Re = 1. For low Reynolds numbers (Re << 1) viscous forces dominate the fluid flow and the “combined” solver should be used, for larger Reynolds numbers (Re > 1) inertial forces become more important and the “piso” solver should be considered.

Driving File

```
…
# Other numerical parameters
# ==========================
# (*) marks default settings for options
# Selection of the Flow-Solver:
# combined|separated|piso  [verbosity <lvl>]
#     [ steady_[re]start|ana_[re]start [pre_iter <it>] ]
combined verbosity 1
# Time stepping in Flow-Solver:  fix <tstep[s]>
# or:  cfl <CFL-number> [first <tfirst>] [max <tmax[s]>] [min <tmin[s]>]
fix 0.000005
# Convergence criterion for impulse eqn. (absolute) [mue_m/s]
#    (real) [warn(*)|nowarn]
1e-5
# Max. number of iterations for linear solver:
#   (int) [bicgmr2(*)|bicgsafe|bicgstab]
500
# Convergence criterion for pressure eqn. (absolute) [Pa]
#    (real) [warn(*)|nowarn] [adapt(*)|noadapt]
1e-8
# Divergence criterion for continuity equation [1/s]
#    (real) [warn(*)|nowarn]
1E-2
# Max. iterations of the flow solver
50
# Factor for underrelaxation of pressure 0.0 < alpha < 1.0
0.97
# Phase minimum?
…
```

Example 4.5 Numerical parameters for fluid flow

After the solver choice the “steady_start” option can be entered to iterate to a steady fluid dynamics solution before any phasefield calculations. It should be combined with the “pre_iter” option to set the number of flow solver calls and ensure that a steady state has been reached. The “_restart” variant enables iterating to a steady flow after a restart. The “ana_start” option will use an analytical solution for poiseuille or couette flow determined by the boundary conditions (ignoring any solid regions) before applying the iterative solver.

As another optional parameter the flow solver “verbosity” level can be set. This enhances the output of the solver to help track convergence and aid in setting appropriate numerical parameters.

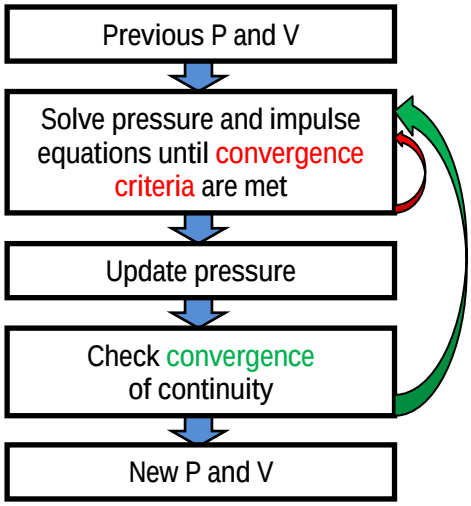

Previous P and V
Solve pressure and impulse equations until convergence criteria are met
Update pressure
Check convergence of continuity
New P and V

Figure 4.26: Flow solver iteration loops

The verbosity level determines at which level of the solver output is produced. Figure 4.26 shows the basic scheme for the flow solver[1]: for each time step the solution is iterated until the continuity criterion is fulfilled, and nested within this loop the impulse- and pressure equations are solved by an iterative linear solver until the convergence criteria for velocity and pressure are met. The different verbosity levels are cumulative and produce output:

1. each flow solver call
2. each continuity iteration
3. additional output
4. each linear solver call
5. each linear solver iteration
6. additional output.

Verbosity is switched off after 100,000 lines of output to prevent clogging the file system. This limit can be changed with the “verbosity_max_lines” option. It is recommended to switch verbosity off once the numerical parameters are adjusted.

After the solver scheme the **time step size** has to be set. It is recommended to use a “fix” time step size. Alternatively the “cfl” option can be chosen. If “cfl” is chosen it should be used in combination with the “max” option to set a maximum step size and/or the “first” option to set the starting time step.

There are two CFL (Courant-Friedrichs-Lewy) numbers that may help to arrive at a good starting point for the time step size, one arising from the advection term in the Navier Stokes equations:

$$C_{\mathrm{adv}} = \frac{\max_{\mathrm{cells}}\left(v_{\mathrm{x}} + v_{\mathrm{y}} + v_{\mathrm{z}}\right) \cdot \Delta t}{\Delta x}$$

where C_{adv} is the CFL number for advection, v_{xyz} are components of the local velocity, Δx is the grid distance of the flow solver and Δt the time step. C_{adv} is relevant for high Reynolds numbers and timesteps can be limited accordingly with the “cfl” option. In that case the piso solver is recommended and a limit $C_{adv} < 0.3$ a conservative choice. Depending on the specific problem higher C_{adv} may work.

For low Reynolds numbers the viscosity term becomes more important and the relevant CFL-number is

$$C_{\mathrm{visc}} = \frac{\nu \cdot \Delta t}{\Delta x^2}$$

with the kinematic viscosity ν. For a given viscosity and grid spacing the timestep is then limited to:

[1] In Figure 4.26 the “combined” solver scheme is shown. For the “separated” and “piso” scheme impulse and pressure equations are solved separately in two linear solvers, and for the piso scheme there is an additional outer loop for impulse (=velocity) convergence.

$$\Delta t_{\max} = \frac{C_{\text{visc}} \cdot \Delta x^2}{\nu}.$$

Here C_{visc} = 0.2 is a conservative choice. In many cases, particularly when flow patterns change slowly with time and simulation cells, much larger timesteps may converge and lead to better performance. Even timestep sizes corresponding to C_{visc} = 250 or C_{adv}=30 have been found to converge in specific cases, but this can not be generalized and it is up to the user to find the optimal time stepping for a particular simulation.

After the time step size convergence criteria and maximal loop counts for the linear solvers need to be set. The convergence criteria limit the absolute rms (root mean square) deviations when solving the impulse- (=velocity-), pressure- or continuity equations. The first criterion limits the velocity error in the impulse equation. When this criterion is failed at the end of a flow solver timestep a warning is produced (after the very first warning only summarized warnings are produced at output timesteps). These warnings can be switched off with the "nowarn" option.

Next the loop count for the linear solver of the impulse equation is queried (in the "combined" case this is the solver for the combined equation system), the default linear solver is "bicgmr2", optionally one may chose "bicgsafe" or "bicgstab". Often iteration in the range 300 – 500 work very well here, but in some cases for large grids higher values (up to 3000) may work better at the start of the iteration (or to converge steady flow patterns). In later stages of a simulation this value should rarely be reached, if it is consider shorter time steps and/or less restrictive convergence criteria.

After that a convergence criterion for the pressure equation is entered. In an adaptive scheme this value is reduced in successive continuity iterations unless the "noadapt" option is used. For the "separated" and "piso" solvers a loop count for the linear solver of the pressure equation needs to be entered, typical limits should be in the range 300 – 500. The "piso" solver also demands an "inner" iteration count for the number of corrector cycles.

Next one needs to enter a criterion for the continuity equation, often 10^{-2} is a good choice here. The unit is 1/s as this limits a "volume flow density". To visualize this, a value of 10^{-2} in one simulation cell over one second would mean that the cells outflow exceeds the inflow by 1% of the cell volume.

The next parameter limits the number of continuity iterations (indicated by the green arrow in Figure 4.26). In some cases higher values (e.g. 20-30) here may help convergence at the beginning of a simulation or a steady flow calculation, but if many iterations are needed in later stages of the simulation, shorter flow solver time steps should be considered.

For the "combined" and "separated" solvers an additional underrelaxation parameter α is needed. For the "separated" solver it is possible to set this value to 1. For the "combined" solver α must be less than 1 and should typically be in the range $0.7 \leq \alpha \leq 0.97$.

4.15.5 Adjusting numerical parameters

For a given fluid flow simulation the errors in velocity, pressure and continuity are correlated, so a matching set of convergence criteria needs to be found. Here it helps to track the convergence by switching on flow solver "verbosity". Example 4.6 shows the convergence of the linear "combined" solver at the beginning of the calculation. The criteria match the input of Example 4.5. The top shows the convergence of the linear solver at verbosity 5, in the combined case pressure- and impulse equation are solved together, so pressure and velocity converge side by side. At the head of the "pressure" and "impulse" column the criteria are shown that the linear solver should reach. These match the parameters from the input file. The 4th and 5th column show the number of "inner" iterations since the last restart and the total iterations in this solver call. Here the linear solver is stopped after 500 cycles and the solution is further converged in subsequent calls to the linear solver. The convergence reached at this stage is shown in the "PV-Residual" line. In the "piso" and "separated" cases the linear solvers for pressure and impulse are separate so either the "impulse" or the "pressure" column is printed. Interspersed between output from the linear solver are output lines resulting from lower verbosity levels.

The bottom shows how to track convergence of the continuity equation with the outer convergence cycles at verbosity 2. With this verbosity the 4th column shows the loop count for continuity and the 5th column the accumu-

lated linear solver iterations. As one can see when comparing the “Final Resolutions” line to the “Criteria” line the pressure criterion is sufficient to reach the desired accuracy for the continuity and the impulse criterion is chosen to match at this point. Here after 28 cycles the continuity criterion (first column) is reached. The high number of iterations in this example is due to the high accelerations at the start of the simulation.

Screen Output

```
…
   PV-solver:                    pressure        impulse iter: inner       total
    Criteria:                  1.00000E-08    1.00000E-05                    500
  PV-Linsolv:                  0.00000E+00    1.27322E+01           1          1
  PV-Linsolv:                  5.39842E-01    2.90440E+02           2          2
  PV-Linsolv:                  2.06884E-02    1.01393E+01           3          3
  PV-Linsolv:                  2.10092E-02    5.94715E+00           4          4
…
  PV-Linsolv:                  6.15636E-07    1.28979E-04          19        497
  PV-Linsolv:                  5.01661E-07    8.82903E-05          20        498
  PV-Linsolv:                  5.01661E-07    8.82903E-05           1        499
  PV-Linsolv:                  8.59733E-07    2.31118E-04           2        500
 PV-Residual:                  7.07738E-07    1.55810E-04          28        500
   Cont. Err:    7.00332E+01    7.07738E-07    1.55810E-04           1        500
…
```

```
…
Flow: t_pf=  1.000000E-04, t_flow=  5.000000E-06, dt=   5.000000E-06
        Type:     continuity       pressure        impulse  iter. cont lin. solver
    Criteria:    1.00000E-02    1.00000E-08    1.00000E-05
   Cont. Err:    7.00332E+01    7.07738E-07    1.55810E-04           1         500
   Cont. Err:    1.91968E+01    9.74374E-06    2.78629E-03           2        1000
   Cont. Err:    9.75127E+00    4.81484E-08    1.76150E-04           3        1500
…
   Cont. Err:    1.54485E-02    7.54372E-09    2.26806E-06          26        9657
   Cont. Err:    1.21605E-02    8.22486E-09    3.38942E-06          27        9886
   Final Res:    9.56403E-03    8.94495E-09    3.65755E-06          28       10067
V_x = 4.3005E+01 V_z = 4.5388E-13 mic/s, CFL-nr = 0.8482E-03
…
```

Example 4.6 Output of linear solver (“combined”): at verbosity 5 (top) and 2 (bottom)

At the beginning of each fluid flow timestep the current times for phasefield (t_pf) and fluid flow simulation (t_flow) as well as the timestepsize (dt) are printed, at the end the velocities averaged over the liquid volume and C_{adv} are printed.

4.15.6 Optimizing flow solver performance

Fluid flow simulations are time intensive, so some care should be taken to optimize them for performance. There are four approaches to speed up your calculation:

- Employing parallelisation, see section 3.2.21.
- Solve fluid flow on a coarser grid, see sections 3.2.3, 4.15.1.
- Optimizing the time step size.
- Optimizing solver type and parameters.

When optimizing time step size and solver parameters it should be noted, that in most simulations it takes some time to establish a steady or near steady state, e.g. a pressure driven flow that accelerates until an equilibrium between accelerating pressure and frictional drag is reached. If you start with a fully liquid domain the “ana_start” option (see section 4.15.4) is a shortcut to establish a steady flow for some boundary conditions. If multiple simulations are done for parameter variations consider starting from a common restart-file. Optimal time step sizes are not necessarily the longest possible time steps, better minimize (linear) solver cycles per simulation time (on

average).

When tuning the convergence parameters (see section 4.15.4) less restrictive convergence conditions and/or fewer cycles benefit performance only up to a point since reduced accuracy in one time step adversely affects performance of subsequent time steps. Also if accuracy is repeatedly reduced in successive time steps this will quickly lead to divergence.

Usually each fluid flow time step is converged until the convergence conditions for continuity and impulse equation are met, but it is possible to use solver loop counts as limiting element. A good example of this is using the “piso” solver with two inner (correction) cycles and only one solver iteration. The convergence of continuity is then governed by the condition for the pressure equation. The error for continuity and impulse equation can then be tracked by the warning messages. When using the solver in this way choose short time steps to ensure convergence.

Besides numerical parameters also the choice of boundary conditions (see section 3.2.19) has a major impact on flow solver performance. Often similar conditions can be modelled with different sets of boundary conditions. Preferable choices of boundary conditions are discussed in section 4.15.3.

If the Boussinesq model is activated and concentration changes are localized consider setting a fixed reference concentration to avoid large scale changes in the pressure field (see section 4.15.2).

5 How to simplify your life

5.1 How to define an initial microstructure

An initial microstructure may be defined as:

- *deterministic:* by determining the amount of grains present at the beginning of the simulation process as well as their exact position and shape. If you choose deterministic grain positioning, you need to define the grain positioning by specifying the X and the Z coordinates of each grain in a cartesian coordinate system with an origin the bottom left-hand corner. As far as the grain shape is concerned, you can define round grains by specifying a radius or rectangular and elliptic grains – by specifying a length axis along the x and the z axes. For the grain distribution in space a Voronoi construction may be used. For more details, refer to chapter 4.6.2 “Voronoi construction“.
- *random:* by defining an integer for randomization. In that case, you need to specify the total number of grains, the number of different types of grains and the number of grains for each grain type. Then, you need to specify the geometry for each grain type – round, elliptic or rectangular. The geometry is defined in the same way as described above. However, in addition you have to specify a minimum and a maximum value for the X-coordinates,. This means that according to the type of geometry you have chosen for your grains, you need to specify a minimum and a maximum radius for the round grains or a minimum and a maximum length along the x-axis for rectangular and elliptic grains. Once all this required information is specified, MICRESS® uses the integer for randomization to position randomly the defined grains across the calculation domain. Usually, “randomization“ is performed to achieve reproducibility of the results. Here, a Voronoi construction can be used as well (check chapter 4.6.2 “Voronoi construction“).
- *from file:* by specifying the exact path of the text file containing information about the grain number, grain positioning and grain shape. The number of grains at the beginning can be either specified explicitly or read from the input file. Alternatively, the grain properties can be input directly in the command line, they can set to “identical“, i.e. all grains have the same properties or they can be divided in blocks, i.e. grain 1 to 3 and grain 4 to 6 if the number of grains was set to 6.
- *no microstructure:* by setting the total number of grains at the beginning to 0. This means that there is no initial microstructure present at the beginning of your process.

For some additional information refer to section “Grain input“ in chapter 3.2.10.

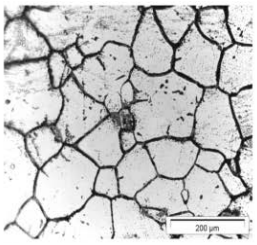
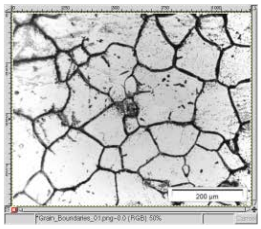
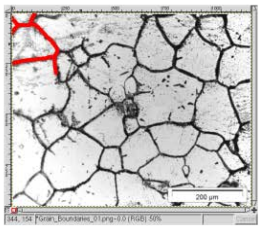
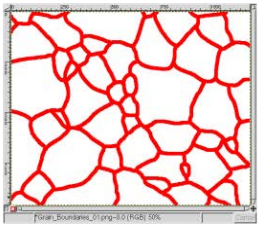
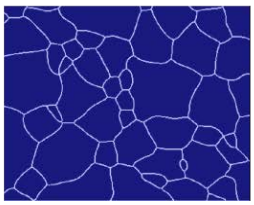
5.2 How to read /include a microstructure file

MICRESS® can read initial microstructure and composition or temperature fields from a file. The file to be read should be an ASCII file with a known “geometry“ (number of cells in X and Y direction): the dimension of the read-in data set does not need to be identical to the simulation one. The most complex case is probably reading an experimental grain structure, but this is not as daunting as it could seem.

Here is a step-by-step description how to do that using GIMP (GNU Image Manipulation Program), Version 1.2.5 (www.gimp.org). The micrograph (the first figure from the top) is courtesy of IEHK, RWTH Aachen. Newer GIMP versions have similar functionalities but the naming might differ. However, any full-fledged image manipulation program should support similar procedures.

- Image/Transform/Zealous Crop/ to get rid of any frame

- Layers/Layers, Channels & Paths/Layers/: insert a new (transparent) layer (second figure from the top)
- Select a clearly different “foreground colour“ (just to help see what has been drawn)
- Tools/Paint → Tools/Pencil to draw the boundaries on the transparent layer.1.Tools/Paint → Tools/Pencil to draw the boundaries on the transparent layer (third figure from the top)
- Layers/Layers, Channels & Paths/Layers/: remove the “background“ layer (original image), figure four from the top). Make sure you do not leave any “dangling“ boundaries, so that each grain is “closed“ (the boundaries of the domain themselves do not have to be re-drawn, as MICRESS will perform this task).
- Image/Mode/Indexed: Use Black/White (1-Bit) Palette, No Colour Dithering Last figure from top to bottom)
- File/Save as/ save to ".pgm", by “exporting“ first. Make sure you save the file in ASCII-code!

For some additional information, check section “Grain input“ in chapter 3.2.10. You can also check the “Grain Growth“ standard example you should have obtained with your installation software.

5.3 How to describe an anisotropic/faceted phase

If a phase shall be defined as anisotropic, you can choose between cubic (which in 2D consists of a 4-fold cosine function) and hexagonal symmetry. Moreover, you shall specify whether the grain orientation shall be defined as a 2D or 3D Euler angle, rotation angles with respect to the axes or Miller indices. The “xyz_axis“ symmetry is a special option for defining anisotropy, where the anisotropy function has the same shape as cubic, but with a preferred direction (i.e. tetragonal).

In the faceted model each facet orientation can be specified explicitly. The faceted orientation requires the specification of all individual normal vectors of the facets in the local coordinate system. It can consist of different types and thus have different properties and can be mirrored further, if necessary, using cubic or hexagonal symmetry. Furthermore, the parameter kappa needs to be specified. This parameter defines the broadness of the anisotropy factor. To view a typical MICRESS® input for faceted growth, click here.

5.4 How to include nucleation

To include nucleation and to obtain additional outputs when nuclei are set or a phase disappears, choose the flag “out_nucleation“ in the beginning of the “Data for further nucleation“ section.

Next, the number of types of seeds should be specified. For each seed type a positioning should be given (bulk, in regions, at the interface, at triple or quadruple junctions). If a nucleation region is requested, you should specify its coordinates (in micrometers). After that, specify the phase state of the new grains and the reference phase. The phase of new grain defines the type of seeds (e.g.: Mg_2Si intermetallics). The “reference phase“ refers to the kind of phase where the new seeds may appear and from which the undercooling is calculated.

If the “interface“ flag has been selected for the grain positioning, it is necessary to define a “substrate phase“. This keyword is used for selecting interfaces between this phase and the matrix phase as nucleation sites, as well as and for taking into account the interface curvature in the undercooling calculation. The substrate phase is the one which determines the extra curvature undercooling and the reference phase is then the one where the undercooling is calculated.

Next, the nucleation model to be used for further nucleation is chosen – the seed density or the seed undercool-

ing model. They have already been presented in chapter 4.7 “Data for further nucleation“, section 4.7.2 “Nucleation models“.

After having chosen the nucleation model to be used in the simulation, MICRESS® asks for a minimum and a maximum nucleation temperature for the different seed types which should lie around the temperature where nucleation is expected.

Noise can be added to the calculation of the driving force for nucleation. In that case, the coefficient for the noise level and a seed for the random number generator will be asked.

One should also specify the maximum number of simultaneous nucleations to be nucleated in one nucleation check.

In the case of solidification of a pure substance (e.g. $T_{melt,Al}$ = 933.55 K) by setting the minimum undercooling to 20 K, one would expect the nuclei to appear at 913.55 K or below. With the help of this flag some stochastic fluctuations are introduced on the undercooling limit.

You can enable the flag “kill_metastable“ to remove metastable grains after their shield time has ellapsed.

An example with $AlSi_{17}Cu_4Mg$ (Figure 5.1) shows how to use the different nucleation modules in a reasonable way:

- *primary silicon Si:* the seed density model/analytical curvature by default
- *Al dendrite:* seed undercooling/stabilisation
- *Si eutectic:* interface nucleation/stabilization
- *Mg2Si:* interface nucleation/stabilization
- *Al2Cu:* interface nucleation/stabilization
- *Al:* interface nucleation/stabilization to achieve complete solidification

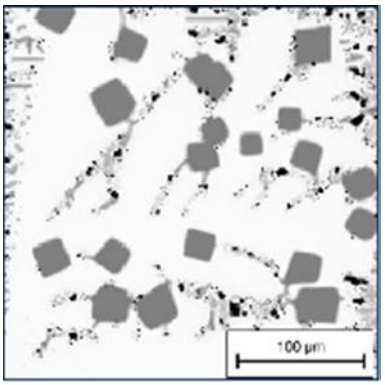


Figure 5.1 AlSi17Cu4Mg

5.5 How to choose between stabilisation and analytical curvature

Stabilisation:

- in the stabilisation model the seed is stabilized by reduction of curvature proportional to phase fraction
- the seed radius must be chosen to be smaller than Δx, otherwise a large grain will be set
- the critical seed undercooling must be smaller than critical undercooling corresponding to a radius of Δx, otherwise the seed may not grow
- use ‘kill_metastable’ to remove such sticking seeds
- use this model with seed_undercooling, if initial growth kinetics are less important than stable growth

Analytical curvature:

- the curvature is calculated analytically (3D) from the phase fraction of the “small grain“, with the largest allowed value corresponding to the critical radius
- if a seed radius larger than [illegible] responds to the largest curvature allowed even after grain increase
- the grain radius is essential for both the curvature and the growth conditions
- mobility is corrected to be as close as possible to real initial kinetics
- use this model for correct description of initial growth or thermal interaction between seeds
- the model is default for the ‘seed-density model’

- the model can also be applied for setting initial grains

5.6 How to include a further alloy element

In order to include a further alloy element, first you need to change (increase) the number of dissolved constituents in the beginning of the “Concentration data“ section. Further input and corrections of the input file depend on whether you use a TC database coupling or a linearised phase diagram.

- For coupling to a thermodynamic database: add the further alloy element in your database. Make sure that when you run the input file through the command line, there is an additional component appearing. Specify the Thermo Calc™ index of the new component. Check whether your matrix phase is the right one. Define how diffusion of the new component shall be resolved for the corresponding phases.
- For linearised phase diagrams: define how diffusion of the new component shall be resolved for the corresponding phases. Then, define the dissolved concentration of the new component in the corresponding phases. Moreover, define the two slopes of the new component relative to the already present components at the reference point. Finally, define the initial concentration of the new component in each of the relevant phases present.

For more information you can refer to some of the MICRESS® examples from the MICRESS® installation directory (e.g. Stress_in.txt, AlCu_in.txt, P_Peak_2D_in.txt).

5.7 How to add a further phase

Consider the Delta_Gamma.dri-example that you have obtained with your installation files. In this example, we will add cementite as a further phase into the simulation. You need to consider the following:

- modifications are necessary at several positions in the .dri-file
- cementite should be considered as isotropic and stoichiometric (Fe3C)
- a cementite spherical grain (e.g. d = 30 m is placed in condition
- the output times should be modified to resolve the dissolution of cementite
- control the traces of cementite in the C-field even after it has dissolved i.e. dissapeared from the .phas-file

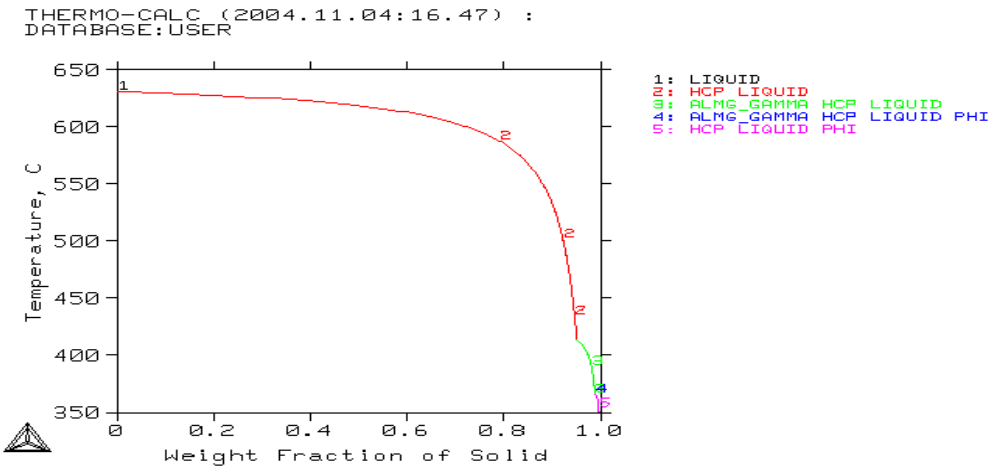


Figure 5.2 A Thermo-Calc Scheil calculation

5.8 How to create .ges-files for TC-coupled simulations (overview)

In order to create a thermodynamic database file, modify and run the create_ges.TCM Macro (see installation directory) using the Thermo Calc™ Classic version.

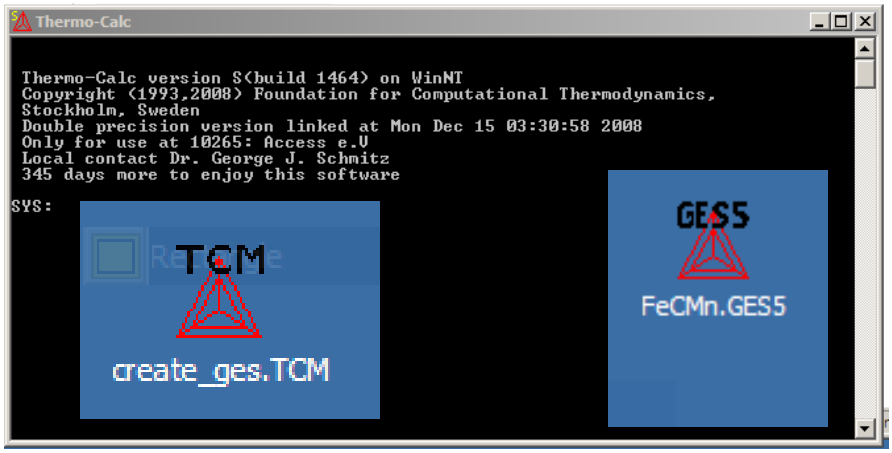


Figure 5.3 Use the TCM delivered along with the MICRESS software as a template to create own ges. files

To begin with, select a thermodynamic database (example: TCFe6 iron and steel database). Then, define the elements in the alloying system and list the system. Next, define the phases of interest. As soon as you are ready with that, get the defined system into the Ges-Workspace and repeat the performed steps. Then, go to the Gibbs workspace and save it to file (example: FeCMn.ges5). For further details, see the next section.

5.9 How to perform a TC-coupled simulation

For simulations with coupling to thermodynamic database, you have to:

- select the relevant components and phases in Thermo Calc™. This means that you should choose important elements which are supported by the database e.g. AZ31: Mg, Al, Zn. Then, you should choose the important phases. For solidification, use the Thermo Calc™ Scheil and for solid-solid reactions – the Thermo Calc™ Equilibrium.
- Create GES5-File from database using Thermo Calc™. The file should contain the same elements as used in MICRESS: MG, AL, ZN. It should also contain at least all phases which will be used in MICRESS®: Liquid, HCP, ALMG_GAMMA, PHI. You should also include mobility data for some phases if available (“append database“ command in Thermo Calc™ Classic)
- Specify all phase interactions. Switch on the most important interactions (e.g. interactions of secondary phases to LIQUID and primary phase (HCP). Leave out complex interactions of minor importance (e.g. ALMG_GAMMA/PHI), especially if stoichiometric phases are involved. Then, the constant-K model is applied (diffusion between phases, but no movement).
- Specify all stoichiometric compositions. Components without solubility range are assigned automatically as stoichiometric. You should also declare stoichiometric all components with restricted solubility range or steep phase diagram slope. Use the “stoichiometric“ option also in case of pseudo-binary “demixing“: $mk\alpha\ \Delta S\alpha * mk\beta\ \Delta S\beta < 0$. In that case, apply the stoichiometric condition to the phase with higher value of $mk\ \Delta S$. The less favourable extrapolation should be compensated by a higher re-linearisation frequency. In case of solidification, an advanced extrapolation scheme can be applied (applies to all definitions after the keyword “stoich_enhanced_on“). Note that it is not possible to define a phase interaction if any component has no solubility range or steep slopes in both phases (versions before release 5.5: if the component is declared stoichiometric in both phases)! Ternary interaction groups (see Section 3.2.14) provide a good alternative to stoichiometric conditions in case of pseudo-binary “demixing“. The “Limits“ keyword can be used to prevent switching between composition sets by setting composition limits in at%.

- Next, specify GES5-File in MICRESS input. Give the path and the name of the .GES5-file for thermodynamic and mobility data. Take care of compatibility (TC version). Increase the workspace size if necessary.
- Specify re-linearisation conditions. All newly formed interface cells request automatically new local linearisations (can be sufficient if the front is moving fast). It is recommended to specify a complete re-linearisation period. A maximum effective temperature deviation can be specified per phase interaction. In that case, re-linearisation is performed for the whole interface between two grains.
- Run MICRESS to get phase and element numbers in database from screen output.

Driving File

```
…
# The database contains the following components:
#   AL
#   MG
#   ZN
# Specify relation between component indices Micress -> TC!
…
# The database contains 4  phases:
#   ALMG_GAMMA
#   HCP
#   LIQUID
#   PHI
# Specify relation between phase indices Micress -> TC!
…
```

- Specify relation to phase and component numbers in MICRESS

Driving File

```
…
# The main component has in MICRESS the index 0
# Thermo-Calc index of (MICRESS) component 0?
2
# Thermo-Calc index of (MICRESS) component 1?
1
# Thermo-Calc index of (MICRESS) component 2?
3
…
# The matrix phase has in MICRESS the index 0
# Thermo-Calc index of the (MICRESS) phase 0?
3
# Thermo-Calc index of the (MICRESS) phase 1?
2
# Thermo-Calc index of the (MICRESS) phase 2?
1
# Thermo-Calc index of the (MICRESS) phase 3?
4
…
```

- Specify the molar volumes of the phases
- Specify a temperature for initial (quasi-) equilibrium. In most cases, the initial temperature is a good choice.
- Check the initial linearization parameters in the .log-file. There should be no extreme values (stoichiometric components?). For stoichiometric phases without solubility range, -999.99 will be displayed as a slope. Furthermore, check whether there is any de-mixing ($mk\alpha\ \Delta S\alpha * mk\beta\ \Delta S\beta < 0$). Make sure that the

initial components are reasonable. If a phase is not present at the beginning, linearization parameters will be calculated when nucleation is checked for the first time.

Hints for troubleshooting:

- most numerical problems in MICRESS® make first Thermo Calc™ complain! Check mobilities (start always with small values, check .driv output), resolution, etc
- make sure to declare properly all stoichiometric components! Check the initial linearization output at initialisation or before first nucleation of a new phase (.log-file). The initialisation of the phase equilibria can be controlled by the "temperature for initial equilibrium" or by manual input of initial compositions
- watch out for de-mixing effects (different slopes in solidus and solvus) or slopes near 0. Defining suitable phases/components as stoichiometric can prevent numerical problems at the cost of a more frequent re-linearisation
- check interface per interface by deactivating the other ones (no_phase_interaction)
- use "no_phase_interaction" if no driving force can be calculated (e.g. if both phases have no or a very small solubility range for the same component or if linearisation fails). Then, the interface will not move but allow diffusion between the phases
- in case of a miscibility gap, separate phase description into composition sets (using Thermo Calc™), which can have different phase numbers in MICRESS®
- extremely small solubility (e.g. N in fcc at lower temperatures) can cause numerical accuracy problems! Use a double precision MICRESS® version and a small value for phMin.

5.10 How to go 3D

You can change the simulation from 2D to 3D by modifying the input file. For that purpose, you need to go to the section: “Flags and settings → geometry“ and then add an integer value > 1 for AnzY. This will define the number of cells in the 3rd dimension.

Further modifications are necessary for the grain orientations i.e. switching from “angle 2D“ to a suitable 3D description.

Also the description of the positions of all grains being set deterministically to define an initial microstructure have to be complemented by the information about the 3rd dimension.

Further, all boundary conditions have to be altered to the respective 3D descriptions.

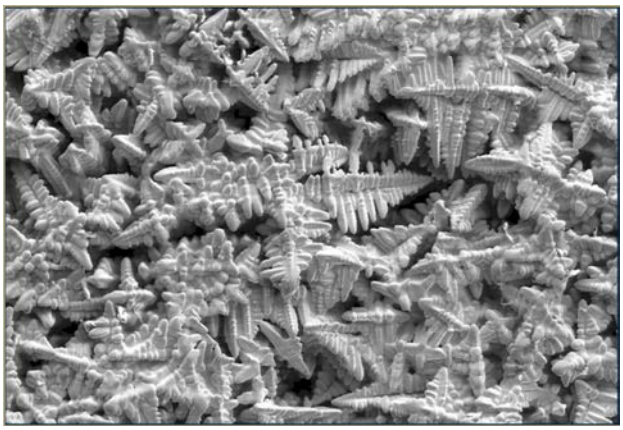


Figure 5.4 Experimental 3D dendrites prepared by removing the melt by centrifugal force

You should check whether the simulation runs. If you not wish to overwrite the existing input file, you should change the corresponding settings under the “Restart options“.

5.11 How to understand and model solidification

Solidification of metallic alloys is an important application for phase-field simulations. The dendritic growth structure is a result of self-organization of the unstable solidification front with a very weak trigger given by the interfacial energy anisotropy. Phase-field simulations capture the underlying physics and make it tractable on a computer. According to the Gibbs–Thomson relation, solidification problems are characterised by:

- the interface velocity coupled to redistribution of heat and solute at the interface
- long-range transport in the bulk phases (Stefan problem)
- interfacial anisotropy is weak and, in good approximation, can be treated as a perturbation of a spherical Wulff shape
- negligible stresses develop on the interface during growth even if solid and liquid differ in density
- heat conduction in both phases is nearly identical
- solute diffusion in solid can be neglected in comparison with solute diffusion in the melt

Issues in dendritic solidification that raise the interest for their modelling at the current stage are:

- the solidification front growing into an undercooled melt is intrinsically unstable and a dendrite can be viewed as a self-organizing structure
- convection in the melt. This is difficult to treat analytically or numerically, but unavoidable under terrestrial conditions
- the analytical solution of steadily growing dendrite approximated by a parabola of revolution is a self-similar solution that does not select the absolute scale, i.e. it fails to predict the scale of the solidification microstructure whereas nature correlates the scale of the microstructure to the material and process conditions well

5.12 How to perform sub-grain modelling

According to the theory, nucleation of recrystallized grains is almost abnormal growth of sub-grains. The input given in Example 5.1 may be helpful:

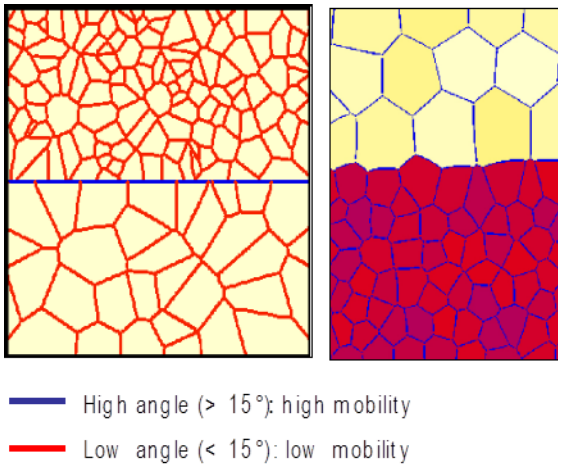


Figure 5.5 Modelling scenario for sub-grain growth using high- and low-angle grain boundaries

Driving File

```
…
# Geometry
...
# Grain input
# ===========
# Type of grain positioning?
# Options: deterministic random from_file
random
# Integer for randomization?
234567
# Number of different types of grains?
2
# Number of grains of type 1?
14
# Number of grains of type 2?
72
# Input for grain type 1
# ----------------------------
# Geometry of grain type 1
# Options: round rectangular elliptic
round
# Minimal value of x-coordinates? [micrometers]
0.
# Maximal value of x-coordinates? [micrometers]
300.
# Minimal value of z-coordinates? [micrometers]
0.
# Maximal value of z-coordinates? [micrometers]
150.
…
```

Example 5.1 Subgrain modeling

5.13 How to handle grains touching the boundaries

Grains touching the boundaries will be “closed“, so that they do not get merged with others. The user is not restricted in the choice of boundary conditions: even wrap-around, though initially leading to somewhat unexpected grain structures, subsequently allow proper relaxation (see the four-folded symmetry view of the “Grain_Growth“ standard example below).

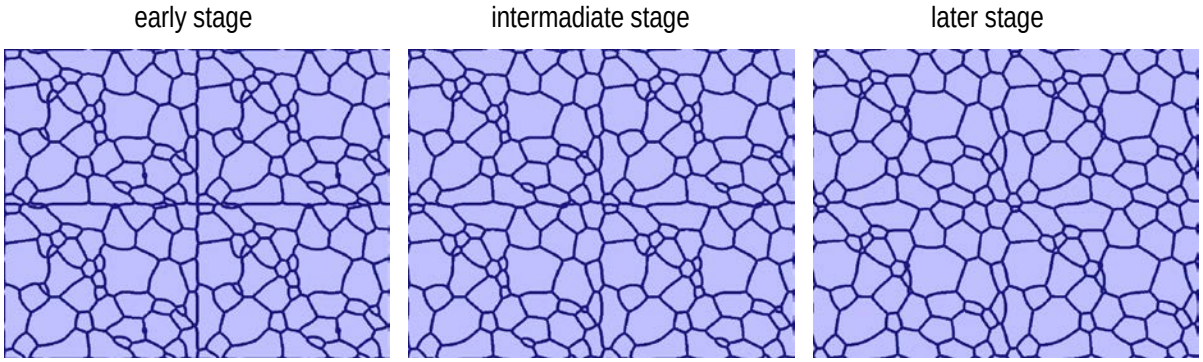


Figure 5.6 A four-folded symmetry view of the "Grain Growth" standard example

6 Literature

[1] B. Böttger, J. Eiken, I. Steinbach. Phase field simulation of equiaxed solidification in technical alloys. Acta materialia. 2006, Vol. 54, pp. 2697-2704.

[2] B. Böttger, J. Eiken, M.Apel. Phase-field simulation of microstructure formation in technical castings – A self-consistent homoenthalpic approach to the micro–macro problem. J. Comput. Phys. 2009, pp. 6784-6795.

[3] W. T. Read, W. Shockley: "Dislocation models of crystal grain boundaries", Phys. Rev., vol. 78, n°3, 1950, pp 275-289.

[4] González, G. M. A phase-field model study of two-dimensional dendritic growth in thermal solidification processes". s.l. : Master Thesis, Supervisor: R. G. Cinca, Departament de Física .

[5] Gupta, S.G. The classical Stefan problem: basic concepts, modelling and analysis. s.l. : Elsevier Science B.N., 2003. 0-444-51085-.

[6] H. Kasajima, E. Nagano, T. Suzuki, S.G. Kim, W.T. Kim. Phase-field Modelling for Faceted Crystal Growth of Silicon. Science and Technology of Advanced Materials. 2003, Vol. 4, pp. 553-57.

[7] H. Meekes, S. X. M. Boerrigter, F. F. A. Hollander, P. Bennema. Needle Crystal Morphology Explained. Chem. Eng. Technol. 2003, Vol. 26, p. 3.

[8] Meirmanov, A. M. The Stefan Problem. De Gruyter Expositions in Math. 3. 1992.

[9] G. Müller, J.-J. Metois and P.Rudolph. Crystal Growth – from Fundamentals to Technology. s.l. : Elsevier B.V., 2004. 0-444-51386-8.

[10] Reed-Hill, R. E. Physical Metallurgy Principles. s.l. : PWS Publishing Company, Boston, 1994. 0-534-98236-0.

[11] Y. Saito, G. Goldbeck-Wood, H. Müller-Krumbhaar. Numerical Simulation of Dendritic Growth. Physical Review A. 1988, Vol. 38, p. 2148.

[12] Schmidt, A. Approximation of Crystalline Dendrite Growth in Two Space Dimensions. Acta Math. Univ. Comenianae. 1998, Vol. LXVII, pp. 57–68.

[13] Stefanescu, D.M. Science and Engineering of casting solidification. s.l. : Kluwer Academic, 2002. 0-306-46750.

[14] Steinbach, I. Phase-field Models in Materials Science. Modelling Simul. Mater. Sci. Eng. 2009, Vol. 17.

[15] Y. Zhang, D. Li, Y. Li. Numerical Simulations of Equiaxed Dendrite Growth Using Phase-field Method. J. Mater. Sci. Technol. 2002, Vol. 18, p. 51.

[16] J. Eiken , B.Böttger, I.Steinbach Multiphase-Field approach for multicomponent alloys with extrapolation scheme for numerical application Phys. Rev. E 73 066122 (2006)

[17] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G. J. Schmitz, J. L. L. Rezende: A phase field concept for multiphase systems, Physica D 94(1996), p.135-147.

[18] J. Eiken, Numerical solution of the phase-field equation with minimized discretization error, IOP Conf. Ser.: Mater. Sci. Eng. 33 012105, doi:10.1088/1757-899X/33/1/012105.

[19] J. Eiken, The Finite Phase-Field Method - A Numerical Diffuse Interface Approach for Microstructure Simulation with Minimized Discretization Error, MRS Proceedings 2012 1369 : mrss11-1369-xx05-05

DOI: 10.1557/opl.2012.510.

[20] S. G. Fries, B. Böttger, J. Eiken, I. Steinbach. S. G. Fries, B. Böttger, J. Eiken, I. Steinbach:"Upgrading CALPHAD to Microstructure: Simulation: the Phase-field Method. Int. J. Mat. Res. 2002, Vols. 100 – 2.

[21] F.J. Humphreys: "A unified theoy of recovery, recrystallization and grain growth, based on the stability and growth of cellular microstructures – I. the basic model", Acta Mater., vol. 45, n° 10, 1997, pp. 4231-4240.